Winter 2-2017

MPLS AND ITS APPLICATION

Ardian Mehmeti

Computer Science and Engineering

MPLS AND ITS APPLICATION

Bachelor Degree

Ardian Mehmeti

February / 2017
Pristine

Computer Science and Engineering

Bachelor Thesis
Academic Year 2010 - 2011

Student: Ardian Mehmeti

MPLS and ITS APPLICATION

Supervisor: Blerton Abazi

February / 2017
Pristine

This thesis is submitted in partial fulfillment of the requirements for a Bachelor Degree

ABSTRACT

Real-time and multimedia applications have grown enormously during the last few years. Such applications require guaranteed bandwidth in a packet switched networks. Moreover, these applications require that the guaranteed bandwidth remains available when a node or a link in the network fails. Multiprotocol Label Switching (MPLS) networks cater to these requirements without compromising scalability. Guaranteed service and protection against failures in an MPLS network requires backup paths to be present in the network. Such backup paths are computed and installed at the same time a primary is provisioned. This thesis explains the single-layer restoration routing by placing primary as well as backup paths in MPLS networks. Our focus will be on computing and establishing backup paths, and bandwidth sharing along such backup paths. We will start by providing a quick overview of MPLS routing. We will identify the elements and quantities that are significant to the understanding of MPLS restoration routing. To this end, we will introduce the information locally stored at MPLS nodes and information propagated through routing protocols, in order to assist in efficient restoration routing. L2VPNs and VPLS will also be covered in the end of this thesis. In the end SDN (software defined networks) will be introduced.

ACKNOWLEGMENT

In this case, I would like to thank my mentor Blerton Abazi who helped me doing this thesis, offering books, different materials and useful advice about the topic that we have chosen and better preparation of this report.

Respect and special thanks to our colleagues which during preparation of these thesis gave us suggestions and criticism, which positively affected the establishment and enrichment of this project with the elements necessary for a proper academic paper.

List of figures

CONTENT

INTRODUCTION

1. History of Mpls

In only a few years, Multi-Protocol Label Switching (MPLS) has evolved from an exotic technology to a mainstream tool used by service providers to create revenue-generating services. There is rapid deployment of MPLS-enabled services and active development of new mechanisms and applications for MPLS in the standards bodies.

The history of MPLS and its precursors is described in [Davie Rekhter] and [Doyle Kolon]. The first Internet Engineering Task Force (IETF) MPLS Working Group Meeting took place in April 1997. That working group still exists, and MPLS has grown to the extent that it underpins much of the activity of several other working groups in the IETF, such as Layer 3 VPN (l3vpn), Layer 2 VPN (l2vpn), PseudoWire Emulation Edge-to-Edge (pwe3). Part of the original MPLS problem statement [MPLS97] from the first MPLS working group meeting is shown below. It contains four items that the group aimed to address through the development of MPLS. It is interesting to examine these to see which items are still relevant today:

- Scalability of network layer routing. Using labels as a means to aggregate forwarding information, while working in the presence of routing hierarchies. Edge routers need to contain routing information pertaining to each VPN that they service, but the core routers do not. Thus, assuming that any edge router services only a subset of the VPNs pertaining to the network, no router in the network needs to hold the entire set of routes present in the network.

- Greater flexibility in delivering routing services. Using labels to identify particular traffic which are to receive special services, e.g. QoS. Using labels to provide forwarding along an explicit path different from the one constructed by destination-based forwarding. MPLS has the ability to identify particular traffic flows which must receive special services such as Quality-of-Service (QoS). It also has traffic engineering properties that allow it to provide forwarding along a particular explicit path. These two properties are combined in DiffServ Aware Traffic Engineering.

- Increased performance. Using the label-swapping paradigm to optimize network performance. Because modern routers perform packet forwarding in hardware, the forwarding rates for IP and MPLS packets are similar. However, optimizing network performance' implies a wider context than simply the performance of individual nodes. Certainly MPLS has helped in this wider context, e.g. through the use of traffic engineering to avoid congestion and the use of fast reroute to reduce the interruption to traffic when a link in the network fails.

- Simplify integration of routers with cell switching based technologies: making cell switches behave as peers to routers (thus reducing the number of routing peers that a router has to maintain), b) by making information about physical topology available to Network Layer routing procedures, and c) by employing common addressing, routing, and management procedures.

1.1 Pre-MPLS Protocols

Before MPLS, the most popular WAN protocols were ATM and Frame Relay. Cost-effective WAN networks were built to carry various protocols. With the popularity of the Internet, IP became the most popular protocol. IP was everywhere. VPNs were created over these WAN protocols. Customers leased ATM links and Frame Relay links or used leased lines and built their own private network over it. Because the routers of the provider supplied a Layer 2 service toward the Layer 3 customer routers, the separation and isolation between different customer networks were guaranteed. These kinds of networks are referred to as overlay networks. Overlay networks are still used today, but many customers are now using the MPLS VPN service. The next section details the benefits of MPLS. It will help you understand why MPLS is a great benefit to the service providers that deploy it and to their customers.

The MPLS labels are advertised between routers so that they can build a label-to-label mapping. These labels are attached to the IP packets, enabling the routers to forward the traffic by looking at the label and not the destination IP address. The packets are forwarded by label switching instead.

When a router forwards an IP packet, it does not change a value that pertains to the destination of the packet; that is, it does not change the destination IP address of the packet. The fact that the MPLS labels are used to forward the packets and no longer the destination IP address have led to the popularity of MPLS.

1.2 Benefits of MPLS

This section explains briefly the benefits of running MPLS in your network. These benefits include the following:

- The use of one unified network infrastructure
- Border Gateway Protocol (BGP)-free core
- The peer-to-peer model for MPLS VPN
- Optimal traffic flow
- Traffic engineering

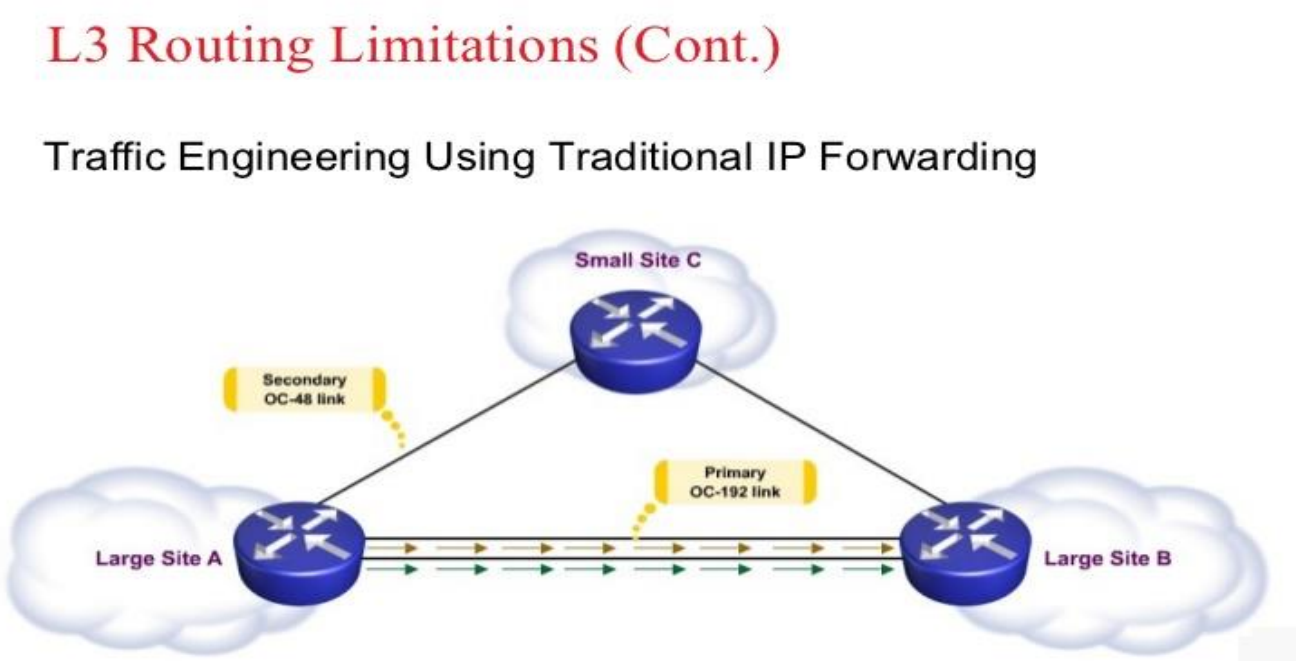
Consider first a bogus reason to run MPLS. This is a reason that might look reasonable initially, but it is not a good reason to deploy MPLS.

One of the early reasons for a label-swapping protocol was the need for speed. Switching IP packets on a CPU was considered to be slower than switching labeled packets by looking up just the label on top of a packet. A router forwards an IP packet by looking up the destination IP address in the IP header and finding the best match in the routing table. This lookup depends on the implementation of the specific vendor of that router. However, because IP addresses can be unicast or multicast and have four octets, the lookup can be complex. A complex lookup means that a forwarding decision for an IP packet can take some time. Although some people thought that looking up a simple label value in a table rather than looking up the IP address would be a faster way of switching packets, the progress made in switching IP packets in hardware made this argument a moot one. These days, the links on routers can have a bandwidth on interfaces up to 100 Gbps. A router that has several high-speed links would not be able to switch all the IP packets just by using the CPU to make the forwarding decision. The CPU exists mainly to handle the control plane.

1.3 Traditional routing

Before explaining basic MPLS functionality, three drawbacks of traditional IP forwarding should be highlighted:

- Routing protocols are used on all devices to distribute the routing information.
- Regardless of the routing protocol, routers always forward packets based on the destination address only. The only exception is policy-based routing (PBR) that bypasses the destination-based routing lookup.
- Routing lookups are performed on every router. Each router in the network makes an independent decision when forwarding packets.



L3 Routing Limitations (Cont.)

Traffic Engineering Using Traditional IP Forwarding

Fig 1. Traditional routing example

MPLS helps reduce the number of routing lookups, possibly changes the forwarding criteria, and eliminates the need to run a particular routing protocol on all the devices.

2.Terminology

The path created in an MPLS network is called a label switched path. Each MPLS enabled router in the network is considered a label switching router. Finally, the actual forwarding of packets is accomplished using a header value that contains a numeric label value. Let's take a closer look at what these terms actually mean.

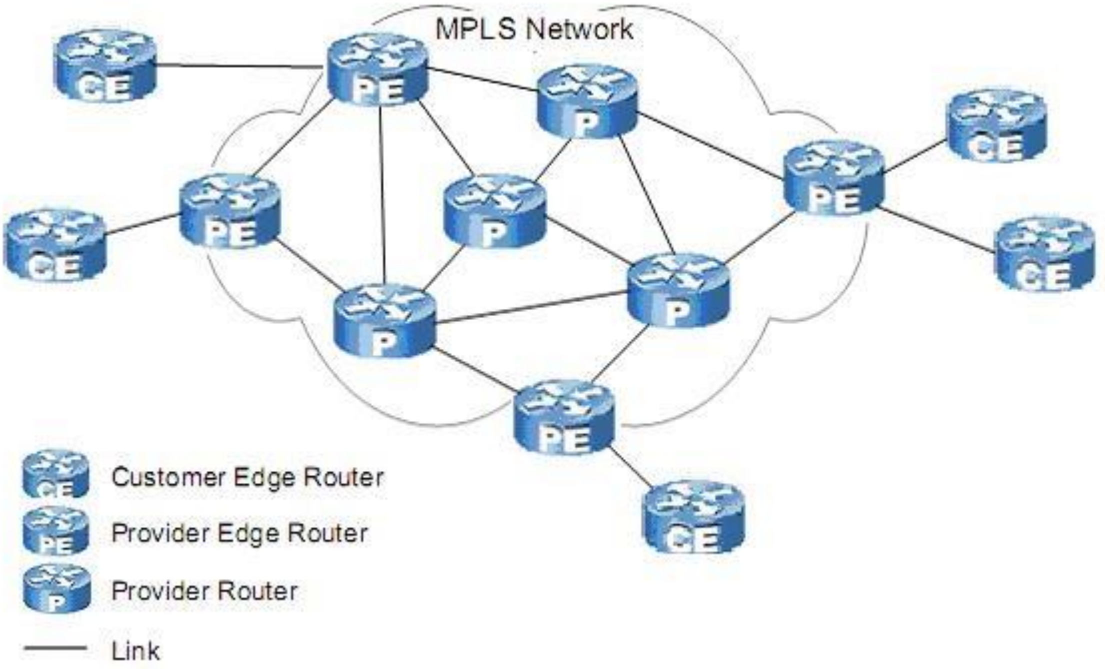


Fig 2. Routers role In MPLS cloud

2.1Forwarding Equivalence Class (FEC)

A FEC is a group/flow of packets that are forwarded along the same path and treating with the same with regards to forwarding treatment. All packets belonging to the same FEC have the same label. However not all the packets that have the same label belonging to the same FEC because their forwarding treatment could be different and they could belong to the different FEC. The router which decides which packets belong to which FEC is Ingress LSR. We can consider few examples:

- Packets with layer 3 destination IP addresses matching a certain prefix (IP prefix/host address)
- Multicast packets belonging to certain group
- Layer 2 circuits (ATM, FR, PPP, HDLC, Ethernet)
- Layer 2 frames carried across an MPLS network received on one VC or sub interface on the
- Ingress LSR and transmitted on one VC or sub interface on the Egress LSR.

- Packets with layer 3 IP addresses that belongs to set of BGP prefixes, all with the same BGP next hop.
- Tunnel interface – traffic engineering

2.2 Label Switched Path (LSP)

Each network path created by the MPLS protocol is a label switched path (LSP). This path is a unidirectional entity that typically exists within a single autonomous system or domain. This one-way traffic flow is different from that of many ATM VCs, which are usually established in a bidirectional manner. The use of a unidirectional system allows you ultimate control of your traffic but does require LSPs to be established in both transmit and receive directions for total traffic engineering in the network.

2.3 Label Switching Routers (LSR)

Each IP router that supports the MPLS protocol is called a label switching router (LSR). An LSR understands the MPLS header and the values encoded within it. The LSR is also responsible for the actual forwarding of user data traffic through the established LSP. There are four different types of LSRs: ingress, transit, penultimate, and egress.

2.4 Ingres Router

The ingress router in an LSP is the only entry point for user data traffic into MPLS. Native IPv4 packets are encapsulated into the MPLS protocol at this location by way of a label push operation. Once encapsulated, packets flow to the egress of the LSP in a downstream fashion. Hence, the ingress router is upstream from the perspective of the data flow. Each LSP in a network must have an ingress router. In addition, only a single ingress router may exist per LSP.

2.5 Transit Router

All routers located in the middle of an LSP are considered transit routers. An individual path can contain between 0 and 253 such routers. The function of a transit router is quite simple. The router checks all received MPLS packets for an incoming label value, which it then looks up in an MPLS forwarding table. After locating the label, the transit router performs a label swap operation by replacing the incoming label with an outgoing label value and decrements the MPLS TTL by 1. The router then forwards the newly labeled data packet to the next hop of the LSP. This entire operation never utilizes the information in the IP data header.

2.6 Penultimate Router

One of the transit routers in an LSP—the penultimate router—has a special function to perform. This router, which is second to last along the path of the LSP, often performs a label pop operation to remove the MPLS information from the data packet. After consulting the MPLS switching table, the router forwards the resulting data, a native IPv4 packet, to the next hop in the LSP after decrementing the TTL value by 1.

Performing this de-encapsulation function on the penultimate router results in scalability. If we move the de-encapsulation function to the penultimate router, however, the workload of the label pop operation is spread across a greater number of routers. This penultimate hop popping (PHP) system allows an MPLS network to scale to greater proportions.

2.7 MPLS LABEL STRUCTURE

MPLS label is a field of 32 bits with a certain structure

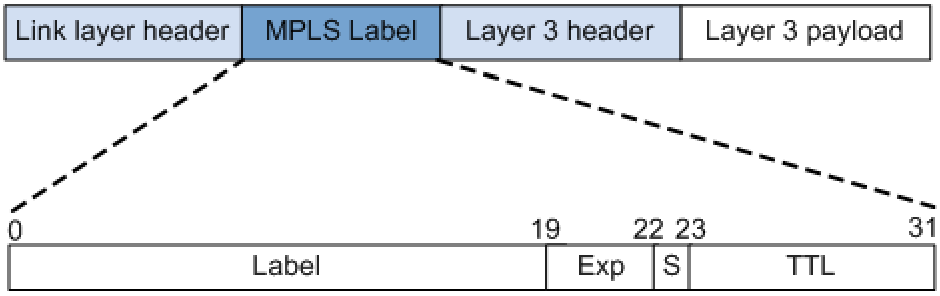
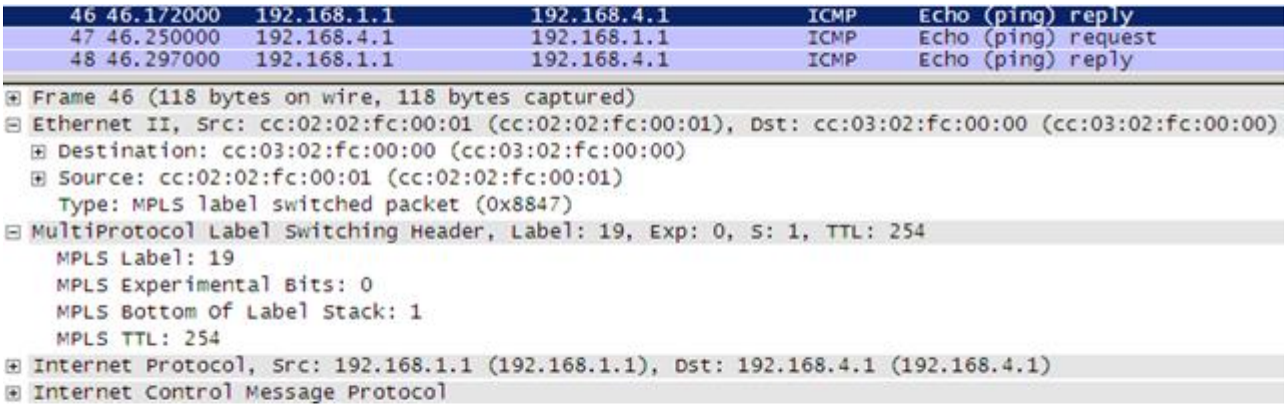


Fig 3. MPLS Label Structure



```
 46 46.172000   192.168.1.1          192.168.4.1          ICMP     Echo (ping) reply
 47 46.250000   192.168.4.1          192.168.1.1          ICMP     Echo (ping) request
 48 46.297000   192.168.1.1          192.168.4.1          ICMP     Echo (ping) reply
⊞ Frame 46 (118 bytes on wire, 118 bytes captured)
⊟ Ethernet II, Src: cc:02:02:fc:00:01 (cc:02:02:fc:00:01), Dst: cc:03:02:fc:00:00 (cc:03:02:fc:00:00)
  ⊞ Destination: cc:03:02:fc:00:00 (cc:03:02:fc:00:00)
  ⊞ Source: cc:02:02:fc:00:01 (cc:02:02:fc:00:01)
    Type: MPLS label switched packet (0x8847)
⊟ MultiProtocol Label Switching Header, Label: 19, Exp: 0, S: 1, TTL: 254
    MPLS Label: 19
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 254
⊞ Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.4.1 (192.168.4.1)
⊞ Internet Control Message Protocol
```

Fig 4. Packet capture via Wireshark showing the MPLS label

3. LSP Signaling Protocols

There are four protocols that can perform the label distribution function:

- Label Distribution Protocol (LDP)
- Resource Reservation Protocol with Traffic Engineering Extensions (RSVP-TE)
- Multiprotocol BGP

LDP and RSVP-TE are the two most commonly used label distribution protocols

3.1 LDP

The Label Distribution Protocol (LDP) is used to establish MPLS transport LSPs when traffic engineering is not required. It establishes LSPs that follow the existing IP routing table, and is particularly well suited for establishing a full mesh of LSPs between all of the routers on the network.

LDP can operate in many modes to suit different requirements; however the most common usage is unsolicited mode, which sets up a full mesh of tunnels between routers.

- In solicited mode, the ingress router sends an LDP label request to the next hop router, as determined from its IP routing table. This request is forwarded on through the network hop-by-hop by each router. Once the request reaches the egress router, a return message is generated. This message confirms the LSP and tells each router the label mapping to use on each link for that LSP.
- In unsolicited mode, the egress routers broadcast label mappings for each external link to all of their neighbors. These broadcasts are fanned across every link through the network until they reach the ingress routers. Across each hop, they inform the upstream router of the label mapping to use for each external link, and by flooding the network they establish LSPs between all of the external links.

The main advantage of LDP over RSVP is the ease of setting up a full mesh of tunnels using unsolicited mode, so it is most often used in this mode to set up the underlying mesh of tunnels needed by Layer 2 and Layer 3 VPNs. In a network, traffic engineering is the ability to control how packets get from one edge of the network to the other.

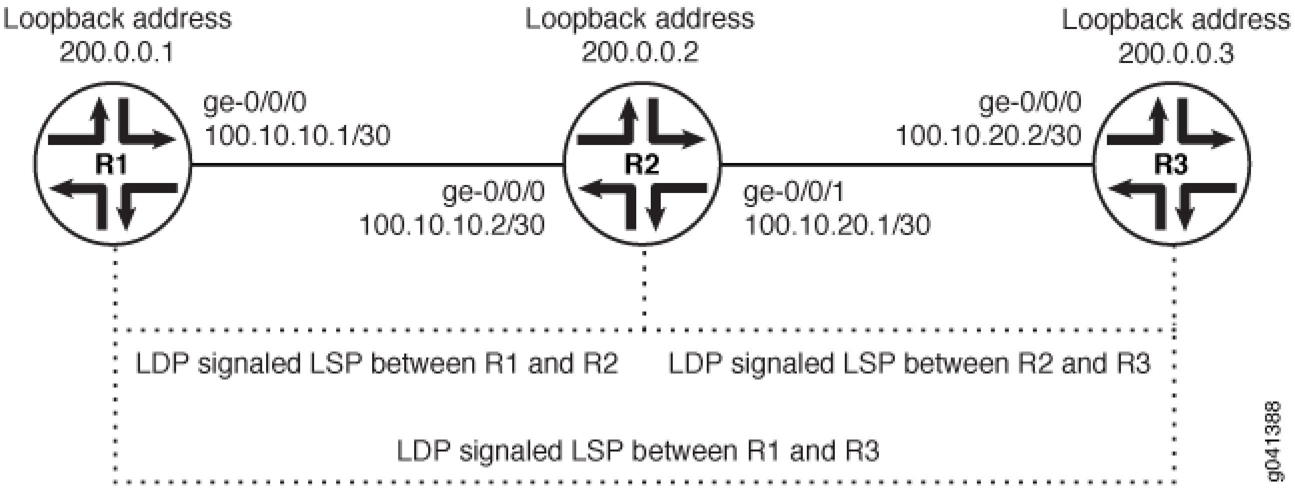


Fig 5. LDP signaled LSP’s

3.1 LDP neighbor discovery and session establishment

First the neighbors discover each other as LDP neighbors via one of two methods (this circumvents the need for manually configuring LDP neighbors):

- **Basic Discovery Mechanism –** Using multicast UDP hellos in the case of direct connected neighbors.

- **Extended Discovery Mechanism –** Using targeted UDP hellos in the case of non-directly connected neighbors.

NOTE In both cases the traffic is destined to the LDP well-known port number 646.

The exchange of LDP Discovery Hellos between two LSRs triggers LDP session establishment, which is a twostep process:

Transport connection establishment (The TCP session using the well know port number 646 – Client-Server TCP operation).

During this process we have two probabilities:

- If the two LSRs already had a TCP session between each other (an already established LDP session over another interface), thus it won't create a new TCP session.

- If the two LSRs had no established TCP session between each other, thus they attempt to open a new TCP connection, and they decide which of them takes the active (acting as the TCP session client using a random source port) and which takes the passive role (acting as the TCP session server listening on the well-known LDP port 646) by comparing the transport address (exchanged in the discovery hellos), and the LSR with the higher address plays the active role and the other plays the passive role.

3.2 Session initialization

After the LSRs establish a transport connection they negotiate the session parameters by exchanging LDP Initialization messages, the parameters negotiated include LDP protocol version, label distribution method, timer values, etc. After the negotiation is successful the LDP session is successfully established.

If the negotiation was not successful (most probably due to incompatible configuration) Error Notification messages are exchanged, and the LSRs retries the session initialization, but since this can result in an endless loop of negotiation, thus an exponential backoff throttling procedure should take effect. After the session is established, now the LSRs can proceed in label distribution. But still there resides a need to maintain the session, which is done on two levels; the maintenance of the Hello Adjacencies (on the discovery level) and the maintenance of the LDP session (on the session level), both are discussed in the following section.

3.3 Hello adjacency and LDP session maintenance

Two LDP peers can have one or more Hello adjacency sessions over multiple links directly connecting between them. Hello messages are used to discover LDP neighbors on each link.

After discovering any LDP neighbor using multicast UDP hello messages, a TCP session must be established for LDP to exchange labels over a reliable connection.

The maintenance of the LDP operation is done on two levels; the hello adjacency level and LDP session level. Both of them are described below

3.4 Maintaining Hello adjacencies

It’s frequent in MPLS networks to see two LSRs connected by multiple links and running label switching over all the links. In this case LDP hello messages are sent on all links with the same LDP identifier. These Hello messages serve two purposes:

1. To auto discover the peers that want to use label switching on this link.
2. To detect peers failures and problems on this link.

If the discovery hold down timer expires without receiving hello messages from the neighbor on one of the links the LSR concludes that the peer no longer wants to run label switching over this link.

When the last hello adjacency (last LDP enabled link between the peers) is deleted (hold down timer expired) the LDP session is terminated by sending a notification message and closing the transport connection.

3.5 Maintaining LDP session

After neighbor discovery, LDP transport session establishment takes place using TCP. LDP maintains this session by sending/receiving keepalive messages and using a hold down timer. Every time an LDP message is received over the session the timer is reset. If the hold down timer expired without receiving LDP messages or keepalives from the peer the transport session is terminated.

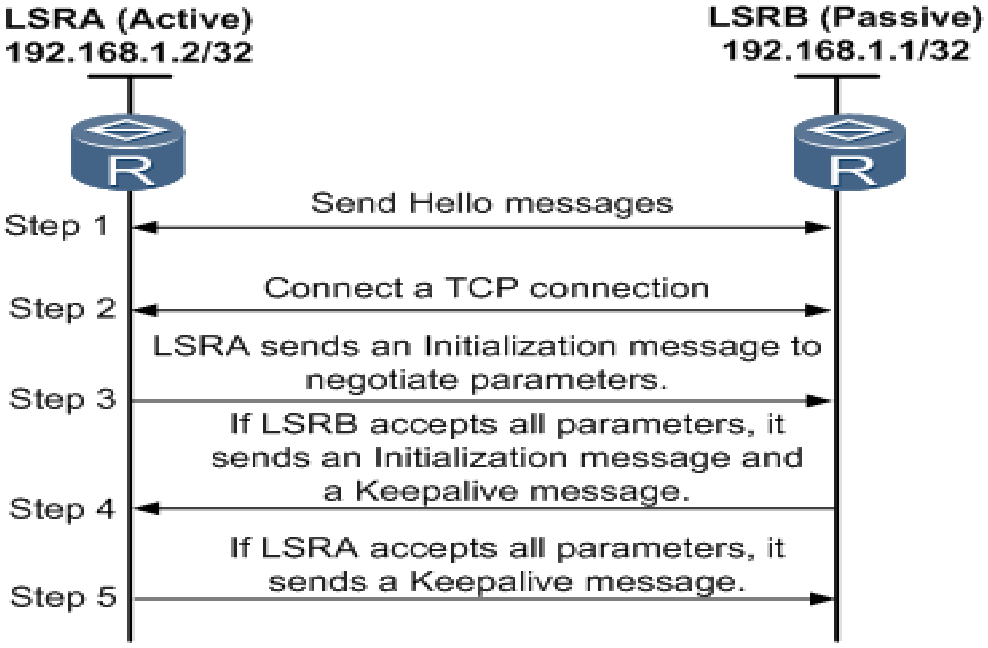

LSRA (Active)
192.168.1.2/32
LSRB (Passive)
192.168.1.1/32
Step 1
Send Hello messages
Step 2
Connect a TCP connection
Step 3
LSRA sends an Initialization message to negotiate parameters.
Step 4
If LSRB accepts all parameters, it sends an Initialization message and a Keepalive message.
Step 5
If LSRA accepts all parameters, it sends a Keepalive message.

Fig 6. LDP neighborship establishment process

Case Study

Part One-Configuration of LDP in the Service provider

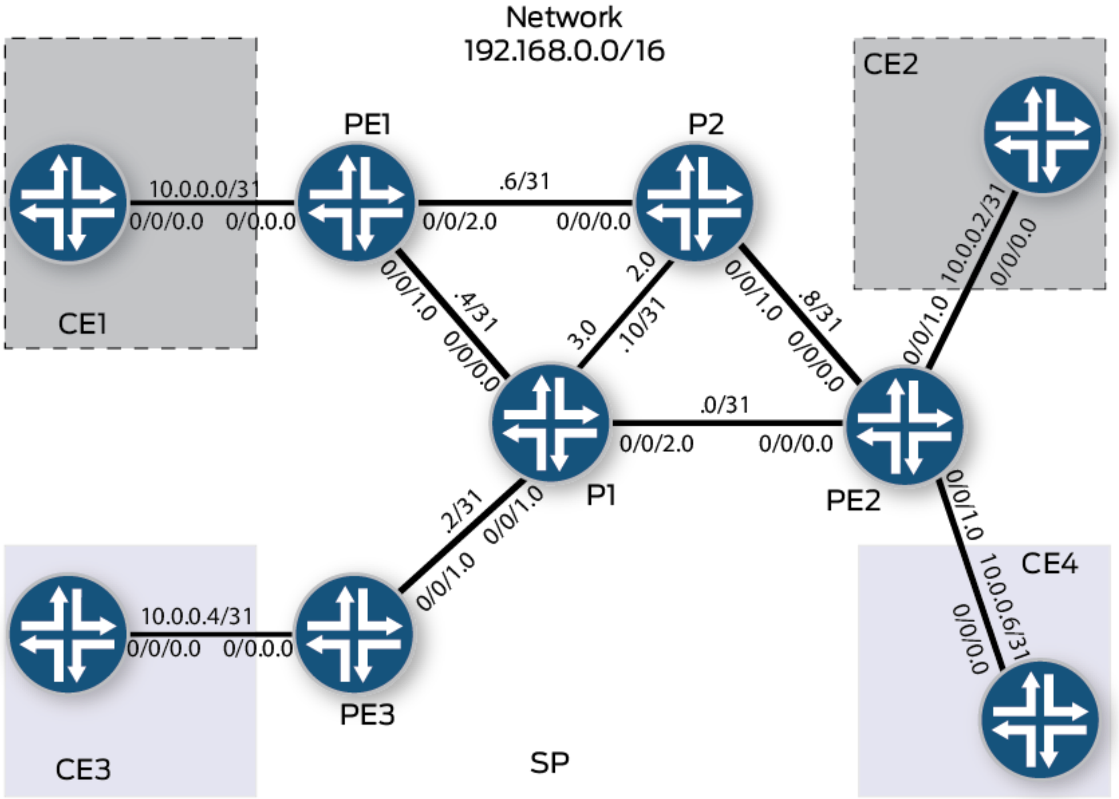


Fig 7. Service provide network topology used for the case study

During the case study Juniper MX routers will be used. Goal of the case study will be to prepare underlying ISP infrastructure for the ISP client's which requested Layer3 and Layer 2 connectivity. Each of the scenario which be shown separately. Also we will show both cases, where LDP and RSVP are used as label distribution Protocol.

For this exercise ISIS is chosen as a routing protocol. Level 2 ISIS will be implemented. On each router *following loopback interfaces will be configured. Also the following ISO addresses will be configured.*

PE1: 172.16.1.1	49.0002.0172.0016.0011.00
PE2:172.16.1.2	49.0002.0172.0016.0012.00
PE3:172.16.1.3	49.0002.0172.0016.0013.00
P1:172.16.1.4	49.0002.0172.0016.0014.00

P2:172.16.1.5   49.0002.0172.0016.0018.00

Configuration for the interfaces facing CE devices will be configured later during Layer2&3 VPN configuration.

PE1 configuration

```
set interface lo0 unit 0 family inet address 172.16.1.1
set interface lo0 unit 0 family iso address 49.0002.0172.0016.0011.00
set interfaces xe-0/0/2 unit 0 family iso
set interfaces xe-0/0/1 unit 0 family iso
set protocols isis level 2 wide-metrics-only
set protocols isis level 1 disable
set protocols isis interface xe-0/0/1.0 point-to-point
set protocols isis interface xe-0/0/2.0 point-to-point
```

MPLS and LDP configuration:

```
set interfaces xe-0/0/1.0 family mpls
set interfaces xe-0/0/2.0 family mpls
set protocols mpls interface xe-0/0/1.0
set protocols mpls interface xe-0/0/2.0
set protocols ldp  interface xe-0/0/1.0
set protocols ldp  interface xe-0/0/2.0
```

PE2 configuration

```
set interface lo0 unit 0 family inet address 172.16.1.2
set interface lo0 unit 0 family iso address 49.0002.0172.0016.0012.00
set interfaces xe-0/0/2 unit 0 family iso
set interfaces xe-0/0/0 unit 0 family iso
```

ISIS configuration:

```
set protocols isis level 2 wide-metrics-only
set protocols isis level 1 disable
set protocols isis interface xe-0/0/0.0 point-to-point
set protocols isis interface xe-0/0/2.0 point-to-point
```

MPLS and LDP configuration:

```
set interfaces xe-0/0/0.0 family mpls
set interfaces xe-0/0/2.0 family mpls
set protocols mpls interface xe-0/0/0.0
set protocols mpls interface xe-0/0/2.0
set protocols ldp interface xe-0/0/0.0
set protocols ldp interface xe-0/0/2.0
```

PE3 configuration

```
set interface lo0 unit 0 family inet address 172.16.1.3
set interface lo0 unit 0 family iso address 49.0002.0172.0016.0013.00
set interfaces xe-0/0/1 unit 0 family iso
```

ISIS configuration:

```
set protocols isis level 2 wide-metrics-only
set protocols isis level 1 disable
set protocols isis interface xe-0/0/1.0 point-to-point
```

MPLS and LDP configuration:

```
set interfaces xe-0/0/1.0 family mpls
set protocols mpls interface xe-0/0/1.0
set protocols ldp interface xe-0/0/1.0
```

P1 configuration

```
set interface lo0 unit 0 family inet address 172.16.1.4
set interface lo0 unit 0 family iso address 49.0002.0172.0016.0014.00
set interfaces xe-0/0/0 unit 0 family iso
set interfaces xe-0/0/1 unit 0 family iso
set interfaces xe-0/0/2 unit 0 family iso
set interfaces xe-0/0/3 unit 0 family iso
set protocols isis level 2 wide-metrics-only
set protocols isis level 1 disable
set protocols isis interface xe-0/0/0.0 point-to-point
set protocols isis interface xe-0/0/1.0 point-to-point
set protocols isis interface xe-0/0/2.0 point-to-point
set protocols isis interface xe-0/0/3.0 point-to-point
```

MPLS and LDP configuration:

```
set interfaces xe-0/0/0.0 family mpls
set interfaces xe-0/0/1.0 family mpls
set interfaces xe-0/0/2.0 family mpls
set interfaces xe-0/0/3.0 family mpls
set protocols mpls interface xe-0/0/0.0
set protocols mpls interface xe-0/0/1.0
set protocols mpls interface xe-0/0/2.0
set protocols mpls interface xe-0/0/3.0
set protocols ldp interface xe-0/0/0.0
set protocols ldp interface xe-0/0/1.0
set protocols ldp interface xe-0/0/2.0
set protocols ldp interface xe-0/0/3.0
```

P2 configuration

```
set interface lo0 unit 0 family inet address 172.16.1.5
set interface lo0 unit 0 family iso address 49.0002.0172.0016.0018.00
set interfaces xe-0/0/0 unit 0 family iso
set interfaces xe-0/0/1 unit 0 family iso
set interfaces xe-0/0/1 unit 0 family iso
set protocols isis level 2 wide-metrics-only
set protocols isis level 1 disable
set protocols isis interface xe-0/0/0.0 point-to-point
set protocols isis interface xe-0/0/1.0 point-to-point
set protocols isis interface xe-0/0/2.0 point-to-point
```

MPLS and LDP configuration:

```
set interfaces xe-0/0/0.0 family mpls
set interfaces xe-0/0/1.0 family mpls
set interfaces xe-0/0/2.0 family mpls
set protocols mpls interface xe-0/0/0.0
set protocols mpls interface xe-0/0/1.0
set protocols mpls interface xe-0/0/2.0
set protocols ldp interface xe-0/0/0.0
set protocols ldp interface xe-0/0/1.0
set protocols ldp interface xe-0/0/2.0
```

Commands used to check if the ISIS and LDP protocols are configure correctly:

```
show isis adjacency
show ldp session
show ldp session detail
labroot@springbok-re0# run show ldp session
  Address           State       Connection     Hold time  Adv. Mode
172.16.1.4         Operational  Open           22         DU
172.16.1.5         Operational  Open           22         DU

[edit]
labroot@PE1-re0# run show ldp session 172.16.1.4 detail
Address: 172.16.1.4, State: Operational, Connection: Open, Hold time: 26
  Session ID: 172.16.1.1:0--172.16.1.4:0
  Next keepalive in 6 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 172.16.1.1, Remote address: 172.16.1.4
  Up for 00:00:04
  Last down 00:00:08 ago; Reason: received notification from peer
  Number of session flaps: 1
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Session flags: none
  Local - Restart: disabled, Helper mode: enabled
  Remote - Restart: disabled, Helper mode: enabled
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Local Label Advertisement mode: Downstream unsolicited
  Remote Label Advertisement mode: Downstream unsolicited
  Negotiated Label Advertisement mode: Downstream unsolicited
  MTU discovery: disabled
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    192.16.0.4
```

3.6 Resource Reservation Protocol

The Resource Reservation Protocol (RSVP) was originally designed to provide end-user hosts with the ability to reserve network resources for data traffic flows. While this concept makes theoretical sense for a single enterprise network, it was never widely implemented for the Internet at large. In essence, the ISPs that make up the Internet didn’t want individual customers altering the operation of their networks.

One of the basic concepts of RSVP is that a traffic flow consists of an identifiable session between two endpoints. Traditionally, these endpoints were the hosts in the network. The concept of a session ties neatly into the concept of an LSP, which transports a traffic flow between two individual routers in the network. This led network designers to extend the RSVP protocol specification to support traffic-engineering capabilities. This extended specification (RSVP-TE) allows an RSVP session to be established between two routers (or endpoints) in the network for the purpose of transporting a specific traffic flow.

Resources are reserved hop by hop across the internetwork; each router receives the resource reservation request, establishes and maintains the necessary state for the data flow (if the requested resources are available), and forwards the resource reservation request to the next router along the path.

RSVP does not transport application data, nor is it a routing protocol. It is simply a label distribution protocol. RSVP uses unicast and multicast IGP routing protocols to discover paths through the internetwork by consulting existing routing tables

3.7. RSVP Basics

RSVP is a signaling protocol that handles bandwidth allocation and true traffic engineering across an MPLS network. Like LDP, RSVP uses discovery messages and advertisements to exchange LSP path information between all hosts. However, RSVP also includes a set of features that control the flow of traffic through an MPLS network. Whereas LDP is restricted to using the configured IGP's shortest path as the transit path through the network, RSVP uses a combination of the Constrained Shortest Path First (CSPF) algorithm and Explicit Route Objects (EROs) to determine how traffic is routed through the network.

RSVP uses unidirectional and simplex (one-way) flows through the network to perform its function. The ingress router initiates an RSVP Path message and sends it downstream to the egress router. This Path message contains information about the requested resources of the connection. Each router along the path begins to maintain a soft state connection for this reservation. You can think of the soft state as a database of current reservations affecting the local router. When the Path message reaches the egress router, the actual reservation of resources begins. This happens with an RSVP Resv message, which is initiated by the egress router and sent upstream to the ingress router. Each router along the path receives the Resv message and sends it upstream, following the route used by the Path message. In addition, more soft state information is added to each local router. Once the ingress router receives the Resv message that matches its original Path message, the unidirectional network path is established. The established network path remains operational as long as the RSVP soft state stays active. This is accomplished through a refresh mechanism where each local router sends Path and Resv messages to its neighbors for all current states every 30 seconds. This informs those neighbors of active paths and assists them in maintaining their own local soft state. The flow of Path and Resv messages in a network is seen in Figure 8.

In addition to the Path and Resv messages, RSVP defines these message types:

PathTear message

The PathTear message always travels downstream to the egress router. It removes the established Path soft state for all routers receiving the message. A transit node sends this message when an outage occurs. The ingress router may also use it when the path is no longer desired.

ResvTear message

The ResvTear message always travels upstream to the ingress router. It removes the established Resv soft state for all routers receiving the message. A transit node sends this message when an outage occurs.

PathErr message

The PathErr message always travels upstream to the ingress router. It denotes an error along the established path. No soft state is removed by routers receiving this message type.

ResvErr message

The ResvErr message always travels downstream to the egress router. It denotes an error along the established path. No soft state is removed by routers receiving this message type.

ResvConf message

The egress router may ask each node along the path for a confirmation that the Resv message was received. The ResvConf message type provides that confirmation message.

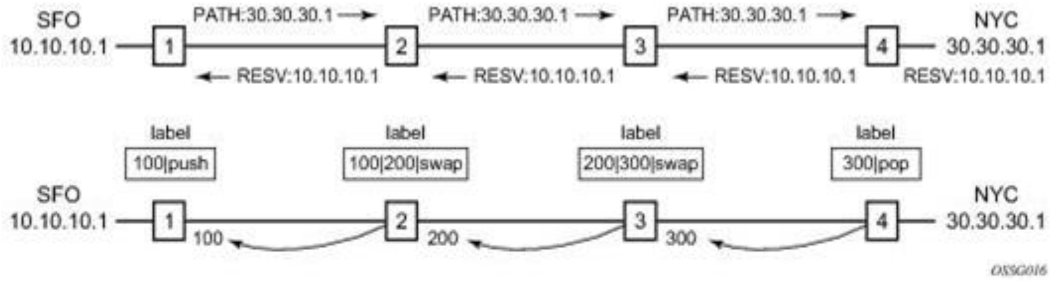


Fig 8. RSVP signaled LSP and label reservation

3.8. Understanding CSPF

CSPF is a link-state algorithm used in computing paths for label-switched paths (LSPs) that are subject to multiple constraints. When computing paths for LSPs, CSPF considers not only the topology of the network, but also the attributes of the LSP and the links, and attempts to minimize congestion by balancing the network load. After pruning paths that do not meet the configured constraints from the shortest-path-first (SPF) tree, CSPF derives the best available path based on the information in the traffic engineering database (TED). Based on the best available path, CSPF produces a strict Explicit Route Object (ERO) which the Resource Reservation Protocol (RSVP) uses to signal the LSP. The CSPF algorithm is a modified version of the SPF algorithm used within the link-state databases of Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) protocols. CSPF operates on the traffic engineering database, which is constructed through extensions to IS-IS and OSPF. Figure 9 illustrates the various components that contribute to the CSPF computation.

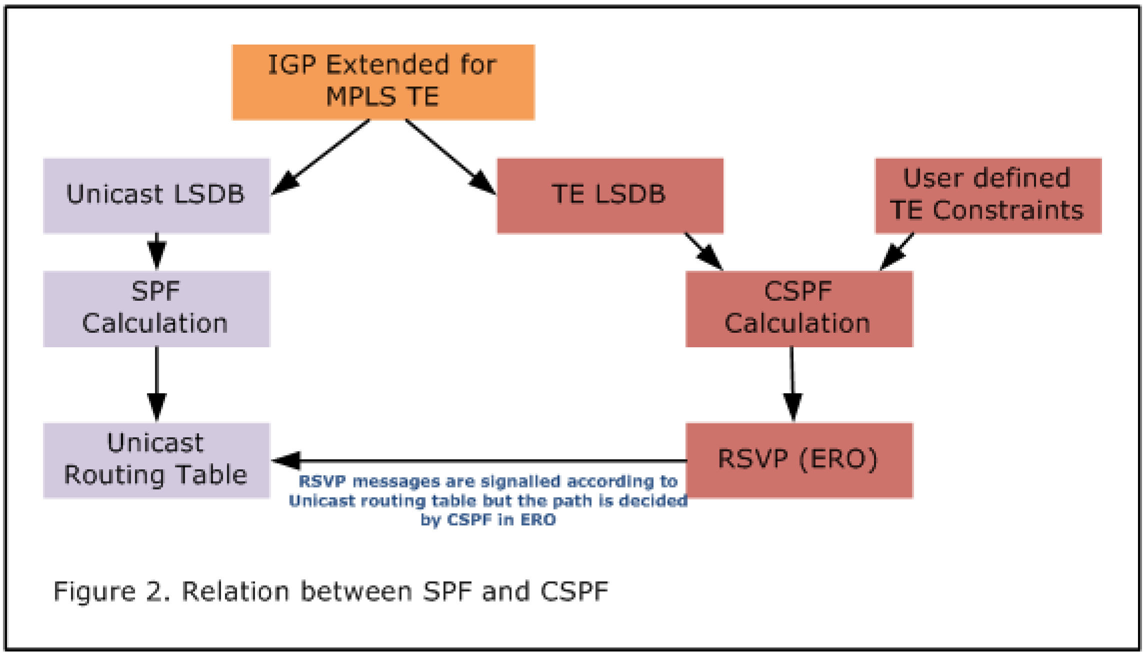

IGP Extended for MPLS TE
Unicast LSDB
TE LSDB
User defined TE Constraints
SPF Calculation
CSPF Calculation
Unicast Routing Table
RSVP (ERO)
RSVP messages are signalled according to Unicast routing table but the path is decided by CSPF in ERO
Figure 2. Relation between SPF and CSPF

Fig 9. IGP extensions for traffic engineering

3.9. IGP Extensions

Both OSPF and IS-IS can propagate additional information through some form of extension. IS-IS carries different parameters in type/length/value (TLV) tuples, which are propagated within a level; these TLVs do not propagate between levels. OSPF, on the other hand, uses Type 10 opaque LSAs to carry traffic engineering extensions. Type 10 LSAs have an area flooding scope, meaning that the information is propagated within a given area only; OSPF traffic engineering extensions do not cross area border routers (ABRs). The MPLS Traffic Engineering Information carried by these IGP extensions is defined in RFCs 3630 and 4203 for OSPF, and RFCs 3784 and 4205 for IS-IS.

3.10. Bandwidth Reservation Requirement

When a bandwidth reservation is configured, reservation messages propagate the bandwidth value throughout the LSP. Routers must reserve the bandwidth specified across the link for the particular LSP. If the total bandwidth reservation exceeds the available bandwidth for a particular LSP segment, the LSP is rerouted through another LSR. If no segments can support the bandwidth reservation, LSP setup fails and the RSVP session is not established.

3.11. Explicit Route Objects

Explicit Route Objects (EROs) limit LSP routing to a specified list of LSRs. By default, RSVP messages follow a path that is determined by the network IGP's shortest path. However, in the presence of a configured ERO, the RSVP messages follow the path specified.

EROs consist of two types of instructions: loose hops and strict hops. When a loose hop is configured, it identifies one or more transit LSRs through which the LSP must be routed. The network IGP determines the exact route from the inbound router to the first loose hop, or from one loose hop to the next. The loose hop specifies only that a particular LSR be included in the LSP.

When a strict hop is configured, it identifies an exact path through which the LSP must be routed. Strict-hop EROs specify the exact order of the routers through which the RSVP messages are sent. You can configure loose-hop and strict-hop EROs simultaneously. In this case, the IGP determines the route between loose hops, and the strict-hop configuration specifies the exact path for particular LSP path segments.

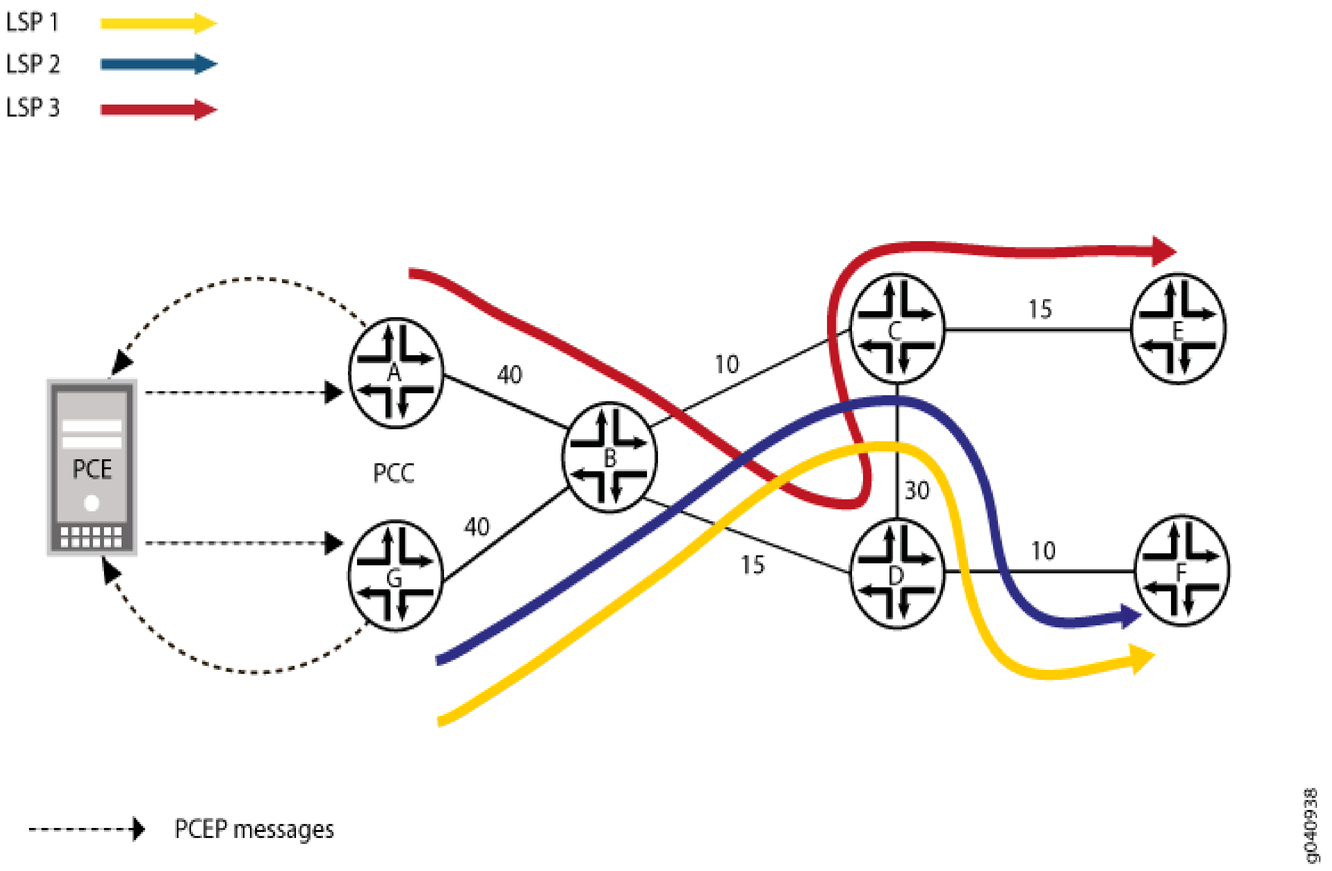


Fig 10. MPLS signaled LSP’s using ERO

4.PROTECTING THE MPLS NETWORK

How the network behaves if a link or a node fails is one of the most important considerations for any network architect and network engineer. Among other things, proper design of failover mode helps ensure that capacity planning can adequately plan and augment the network as required and that quality of service requirements can be met. It is also a major factor in determining how network operations staff respond to a network failure. The failover options discussed here are designed to protect RSVP LSPs against any single point of failure between the ingress and egress routers. Note that if there is more than one failure, the methods below may not work. Additionally, none of the methods discussed here protect against a catastrophic failure of an LSP’s ingress or egress routers.

4.1 Network Failures

When a network failure occurs along the path of an RSVP-signaled LSP, traffic that is currently traversing the LSP will be dropped. In the example, at the instant that the link between R3 and R4 fails, traffic that has already been encapsulated in an MPLS header by R1 and forwarded downstream will be dropped. Also, until R1 receives ResvTear message for the LSP, R1 may continue forwarding traffic using the LSP. That traffic will also be dropped. The time that it takes for traffic flow to be restored depends on the time it takes R1 to be notified of the failure followed as well by the time it takes to resignal a new LSP that will bypass the failed link. There are several features, like fast reroute and link protection which are described in this material that can significantly reduce down time.

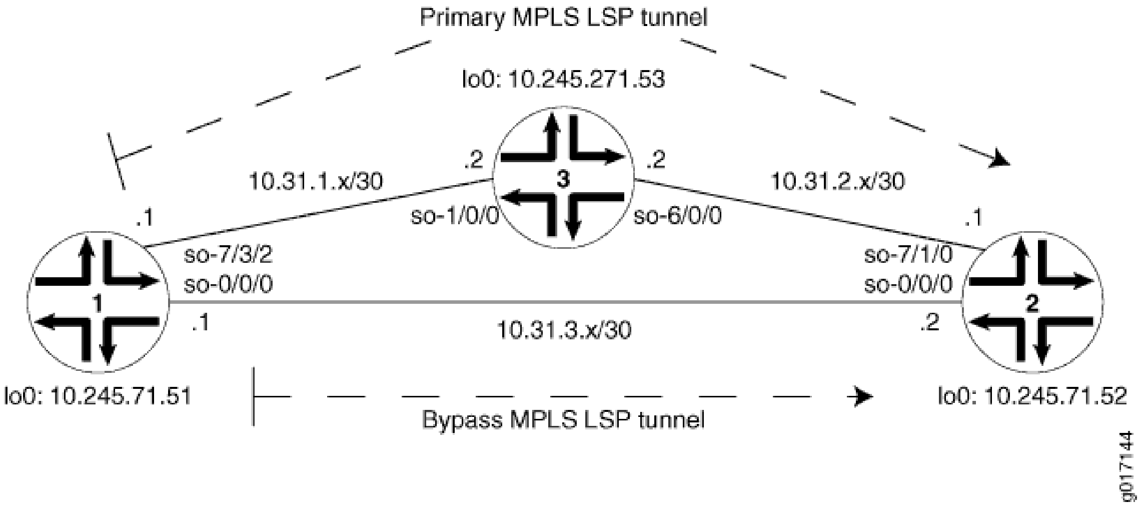


Fig 11. Primary and Bypass signaled LSP example

4.2 RSVP LINK PROTECTION

Link protection provides protection against a link failure along an RSVP label switched path. When link protection is configured, each router along the LSP (except for the egress router) attempts to find an alternate path to the next router in the LSP. This alternate path is known as a next-hop bypass LSP. The next-hop bypass LSP’s purpose is to provide an alternate path to the router on the other side of the protected link. Each next-hop bypass LSP is established after the main LSP is set up. When a link failure along the LSP occurs between two routers, the repair action is initiated by the local router with the failed link that is closest to the LSP ingress router. This router is known as the point of local repair (PLR).

4.3 NODE-LINK PROTECTION

While link protection is useful for selecting an alternate path to the same router when a link fails, node-link protection establishes a bypass LSP through a different router altogether. For Case 1 in Figure 12, link protection allows an LSP to switch to link B and immediately bypass failed link A. However, if Router B fails, link B will fail and the link-protected LSP will be lost.

With node-link protection, the backup LSP can switch to link D instead and bypass the failed links and router. Another benefit of node-link protection shown in Case 2 is that a node-link-protected LSP can act like a link-protected LSP and switch to link B if link D is unavailable.

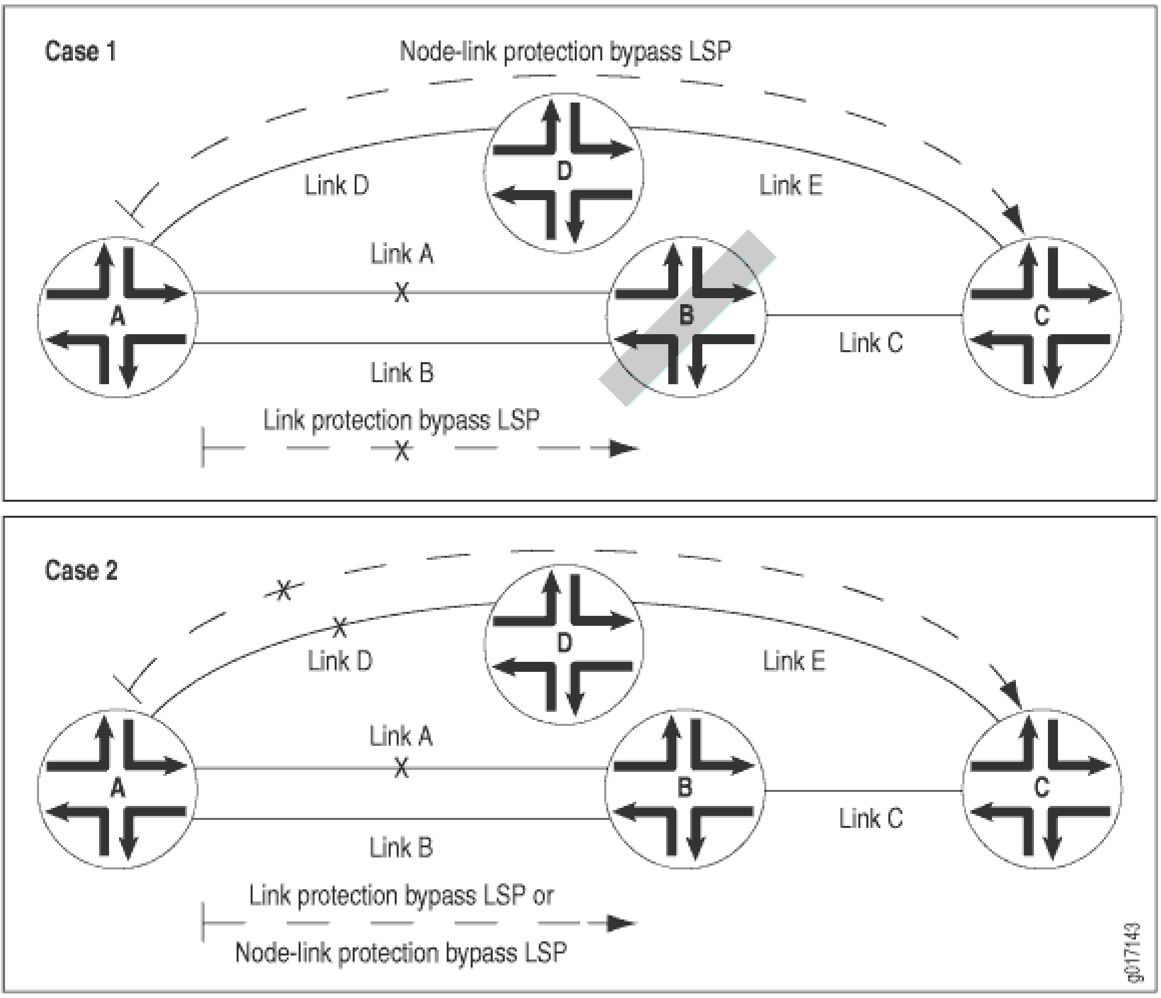

Case 1
Node-link protection bypass LSP
Link D
D
Link E
Link A
A
B
C
Link C
Link B
Link protection bypass LSP
Case 2
Link D
D
Link E
Link A
A
B
C
Link C
Link B
Link protection bypass LSP or
Node-link protection bypass LSP
g017143

Fig 12. Link and node-link protection LSP's

4.4 Fast Reroute

Fast reroute provides redundancy for an LSP path. When you enable fast reroute, detours are precomputed and pre-established along the LSP. In case of a network failure on the current LSP path, traffic is quickly routed to one of the detours. Figure 13 illustrates an LSP from Router A to Router F, showing the established detours. Each detour is established by an upstream node to avoid the link toward the immediate downstream node and the immediate downstream node itself. Each detour might traverse through one or more label-switched routers (or switches) that are not shown in the figure.

Fast reroute protects traffic against any single point of failure between the ingress and egress routers (or switches). If there are multiple failures along an LSP, fast reroute itself might fail. Also, fast reroute does not protect against failure of the ingress or egress routers.

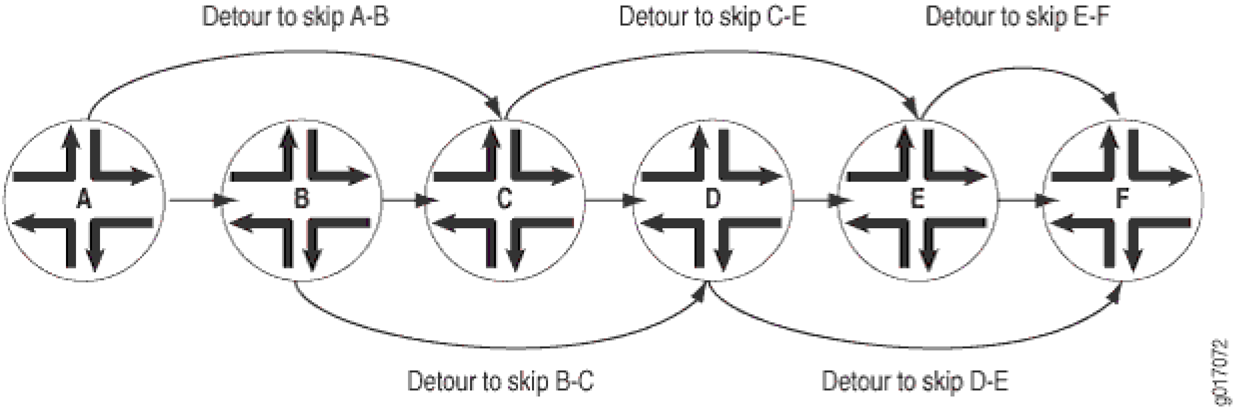


Figure 13. Detours Established for an LSP Using Fast Reroute

If a node detects that a downstream link has failed (using a link-layer-specific liveness detection mechanism) or that a downstream node has failed (for example, using the RSVP neighbor hello protocol), the node quickly switches the traffic to the detour and, at the same time, signals the ingress router about the link or node failure. Figure 13 illustrates the detour taken when the link between Router B and Router C fails.

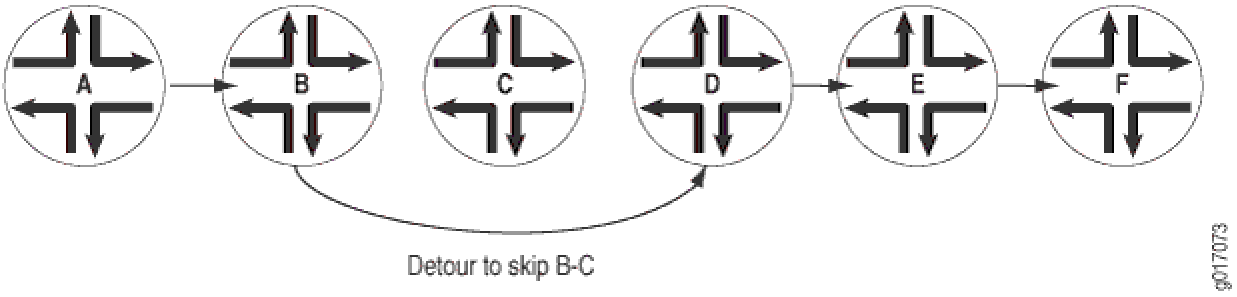


Figure 14. Detour after the Link from Router B to Router C Fails

Case study

Part Two- RSVP, FasteReroute and Link-Node Protection

During Second part of the case study we will remove the ldp as label distribution protocol and We are going to replace it with RSVP. LSPs will be configured between all PE routers.
LSPs are unidirectional so Link protection will be configured for LSPs “PE1-to-PE2”, “PE1-to-PE3”, “PE2-to-PE1” and “PE2-to-PE3”. Fast reroute protection will be configured for LSPs “PE3-to-PE1” and “PE3-to-PE2”.
LDP will be removed on each router by “delete protocol ldp”

PE1 configuration

```
Delete protocol ldp
```

Enabling RSVP on all interfaces:

```
set protocol rsvp interface all
```

LSP configuration:

```
set protocols mpls label-switched-path PE1-to-PE2 to 172.16.1.2
set protocols mpls label-switched-path PE1-to-PE3 to 172.16.1.3
```

PE2 configuration

```
delete protocol ldp
set protocol rsvp interface all
```

LSP configuration:

```
set protocols mpls label-switched-path PE2-to-PE1 to 172.16.1.1
set protocols mpls label-switched-path PE2-to-PE3 to 172.16.1.3
```

PE3 configuration

```
delete protocol ldp
set protocol rsvp interface all
```

LSP configuration:

```
set protocols mpls label-switched-path PE3-to-PE1 to 172.16.1.1
set protocols mpls label-switched-path PE3-to-PE2 to 172.16.1.2
```

```
delete protocol ldp
set protocol rsvp interface all
```

P2 configuration

```
delete protocol ldp
set protocol rsvp interface all
```

LSP between PE1 and PE2 should go via P2 and LSP between PE3 and PE2 should go via P1. We will configure the explicit path which will be signaled by Ingress routers, PE1 respectively PE3.

PE1-configuration and verification before strict path is configured:

```
[edit]

amehmeti@PE1# set protocols mpls label-switched-path PE1-to-PE2 to 172.16.1.2

[edit]
amehmeti@PE1# run show mpls lsp
Ingress LSP: 1 sessions
To           From         State Rt P   ActivePath     LSPname
172.16.1.2   172.16.1.1   Up    0 *                   PE1-to-PE2
Total 1 displayed, Up 1, Down 0


[edit]
amehmeti@PE1# run show mpls lsp name PE1-to-PE2 ingress detail
Ingress LSP: 1 sessions

172.16.1.2
  From: 172.16.1.1, State: Up, ActiveRoute: 0, LSPname: PE1-to-PE2
  ActivePath:  (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                  State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 192.168.0.7 S 192.168.0.8 S
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
20=Node-ID):
          192.168.0.7 192.168.0.8 <<<<<<<<<<<<<<<<signaled path following IGP path
Total 1 displayed, Up 1, Down 0
```

PE1-configuration and verification after strict path is configured:

```
[edit]
amehmeti@PE1# set protocols mpls path PATH-VIA-P1 192.168.0.4 strict

[edit]
amehmeti@PE1# set protocols mpls label-switched-path PE1-to-PE2 primary PATH-VIA-P1

[edit]
amehmeti@PE1# commit
commit complete

[edit]
amehmeti@hoggar:PE1# run show mpls lsp name PE1-to-PE2 ingress detail
Ingress LSP: 1 sessions

172.16.1.2
  From: 172.16.1.1, State: Up, ActiveRoute: 0, LSPname: PE1-to-PE2
  ActivePath: PATH-VIA-P1 (primary)
  LSPtype: Static Configured, Penultimate hop popping
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   PATH-VIA-P1      State: Up
    Priorities: 7 0
```

```
   SmartOptimizeTimer: 180
   Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 20)
 192.168.0.4 S 192.168.0.1 S
   Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
        192.168.0.4 192.168.0.1 <<<<<<<<<<<<path changed and now it’s passing via P1
Total 1 displayed, Up 1, Down 0
```

For configuration of link or link-node protection traffic engineering must be enabled on all routers. When ISIS is used as IGP protocol traffics engineering is enabled by default using TLV-s. In case OSPF is used as IGP traffic engineering must be enabled manually on all routers as below:

```
amehmeti@ PE1# set protocols ospf traffic-engineering
```

5. Layer 3 VPN

5.1 Introduction

BGP/MPLS IP VPNs, referred to in short as MPLS L3VPNs or simply L3VPNs are one of the most widely deployed applications enabled by MPLS. When talking about MPLS, it is not fast reroute or traffic engineering that springs to mind, but rather VPN support. In fact, traffic engineering and fast reroute are most often thought about in terms of the benefits that they can provide in the context of a particular service. Perhaps the most popular service is provider-provisioned IP VPNs and the L3VPN solution described in this thesis is the way this service is realized in MPLS networks. For many providers, L3VPNs is the major and sometimes the only driver for deploying MPLS in the network.

VPNs existed long before MPLS. The success of L3 BGP/MPLS VPNs is owed to the scaling and simplicity advantages that the combination of BGP and MPLS brings to VPN scenarios. The L3 BGP/MPLS VPN solution was extended to the Layer 2 space as well, as we will see in the chapters discussing Layer 2 Transport and VPLS

- Layer 3: the service provider will participate in routing with the customer. The customer will run OSPF, ISIS, BGP, static route or any other routing protocol with the service provider, these routes can be shared with other sites of the customer.
- VPN: routing information from one customer is completely separated from other customers and tunneled over the service provider MPLS network.

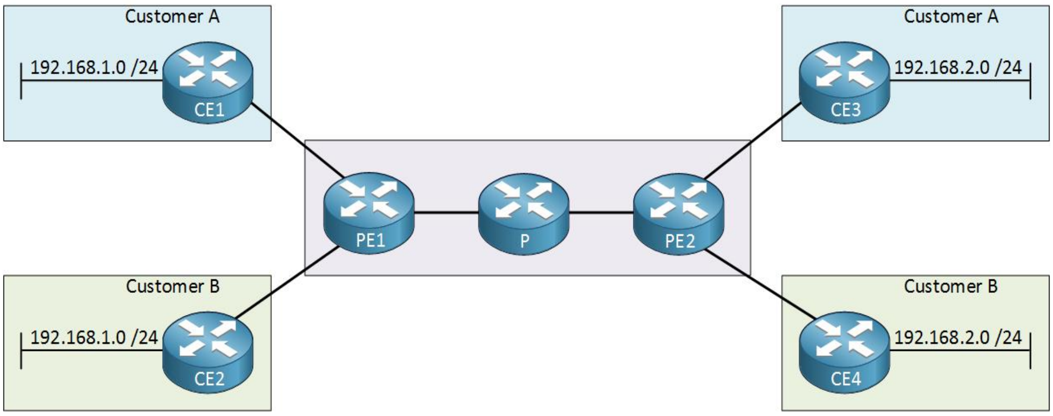


Fig 15. MPLS Layer 3 VPN example with two customers.

Above we have two customers connected to a service provider network. Customer A and B each have two sites and you can see that they are using the same IP ranges.

Customer A might use OSPF between their sites and customer B could use ISIS or RIP between their sites. Everything from these customers is completely separated by the service provider.

Terminology

Provider edge router

The provider edge (PE) router is located in the provider's network and communicates directly with the CE router. In addition, it maintains relationships with other routers inside the provider’s network. For an L3VPN environment, the PE router communicates with its attached CE router to receive routing updates. This information is then advertised to a remote PE router that is connected to another of the customer’s sites. When the PE router receives data packets destined for a remote site, it forwards the packets using a Multiprotocol Label Switching (MPLS) LSP across the provider’s network. When the PE router is participating in a Layer 2 VPN, it simply receives a Layer 2 frame from the local CE router, which it forwards to a remote PE router using an MPLS LSP.

Provider router

The provider (P) router is located within the core of the provider’s network. The P routers do not maintain any knowledge of the customer’s VPN information but simply forward MPLS packets from one PE router to another.

.

VPN forwarding table

A VPN forwarding table (VFT) is used in a Layer 2 VPN environment.
Each PE router creates a separate VFT for each customer connected to that PE router. The VFT contains information that describes the local PE-CE connection, such as encapsulation type, local logical interface, a local site identifier, and some MPLS label information. Each VFT contains knowledge of the remote locations connected across the provider’s LSPs.

VPN routing and forwarding table

This is the first step in separating traffic from different customers. Instead of using a single global routing table, we use multiple routing tables. Each customer of the service provider will use a different VRF/routing-instance. Let’s take a closer look:

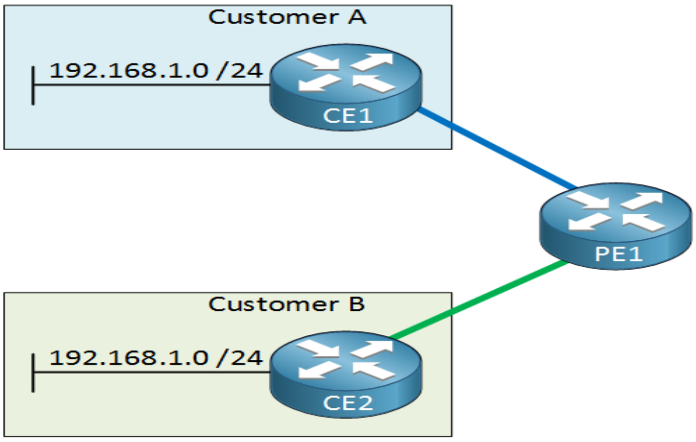


Fig 16. Overlapping IP addresses among two different L3 VPN customer

Above we have our PE1 router with the two customer sites. Each customer will use a different VRF/routing-instance so the overlapping address space is no problem.

5.2 MP-BGP (MultiProtocol BGP)

We will use BGP between the PE routers so that they can share information from the VRFs. Here's how it works:

- One of the CE routers advertises something to the PE router, this can be done through OSPF, EIGRP, BGP or any other routing protocol (static routing is also possible).
- The PE router uses a VRF for the customer so it will store everything it learns in the routing table of the customer's VRF.
- The PE router will then redistribute everything in BGP.
- The PE router will advertise to the other PE router through iBGP.

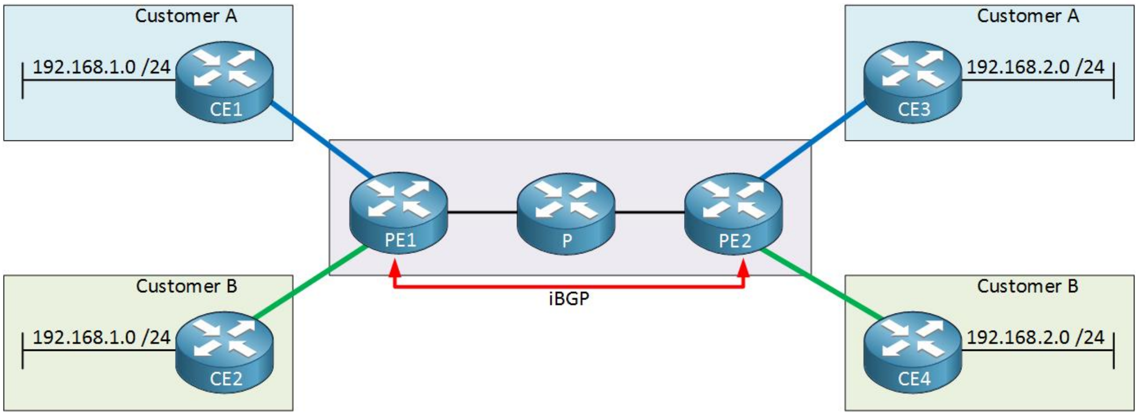


Fig 17. BGP neighborship example between two PE routers

The problem with VRFs is that you have to create them everywhere. When our goal is to have connectivity between CE1 and CE3 then we will have to add a VRF on the PE1, P and PE2 router. Also, all the service provider routes will have to participate with routing. For example, when customer A wants to run OSPF between their two sites then it means that we have to configure OSPF on the PE1, P and PE2 router of the service provider for their VRF.

When customer B wants to run OSPF between their sites, we have to participate…we'll have to configure OSPF on all service provider routers for the VRF of customer B.

This is not a scalable solution so it's not going to happen. Instead, we will configure the VRFs only on the PE routers. The core of the service provider network (P router) will only do switching based on labels.

To share information about VRFs between PE routers, we will use BGP.

There's a couple of problems though. First of all, our two customers are using overlapping address space. Let's say that our PE1 router is advertising 192.168.1.0 /24 from customer A to the PE2 router on the other side. Here's what happens:

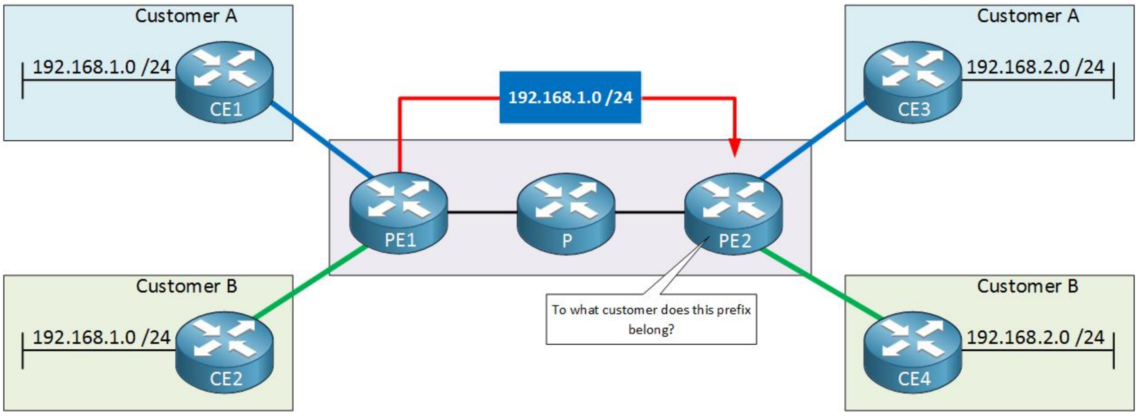


Fig 18.Customer VPN example

The PE2 router will learn 192.168.1.0 /24 from the PE1 router but it has no clue to what customer it will belong. There is no way to differentiate if something belongs to customer A or B.

What we need is something to make all prefixes that we learn **unique**.

5.3 RD (Route Distinguisher)

To fix issue of overlapping address, we will use a **RD (Route Distinguisher)**. We will add something to the prefix of the customer so that it will become unique:

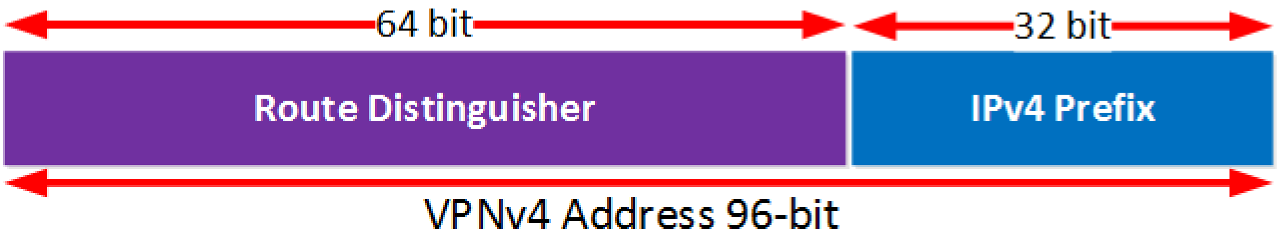


Fig 19. Route distinguisher structure.

The RD is a 8 byte (64 bit) field. You can use any value you want but typically we use the ASN:NN format where ASN is the service provider's AS number and NN is a number we pick that identifies the site of the customer.

The RD and the prefix combined is what we call a **VPNv4 route**. We now have a method to differentiate between the different prefixes of our customers. Here's an example:

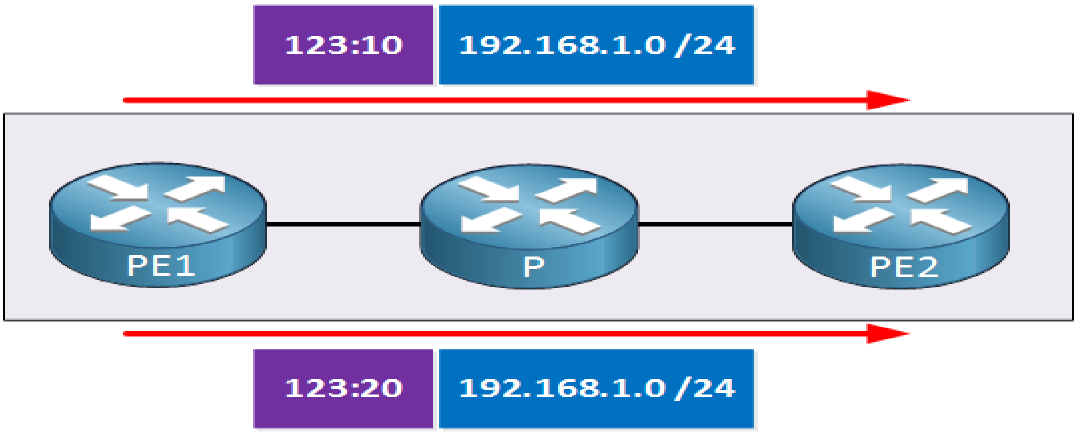


Fig. 20 VPNv4 prefix example

Let's say that we use RD 123:10 for customer A and RD 123:20 for customer B. By adding these values, we have unique VPNv4 routes.

How do we advertise these VPNv4 routes? That's what we need MP-BGP for.

MP-BGP supports IPv4 unicast/multicast, IPv6 unicast/multicast and it has support for VPNv4 routes. To exchange VPNv4 routes, MP-BGP uses a new **NLRI (Network Layer Reachability Information)** format that has the following attributes:

- RD (Route Distinguisher)
- IPv4 prefix
- Next Hop
- VPN Label

This is how PE routers exchange VPNv4 routes with each other. This NRL also has an attribute called the VPN label.

5.4 RT (Route Target)

When a PE router learns these VPNv4 routes, what will it do with it? Take a look at the picture below:

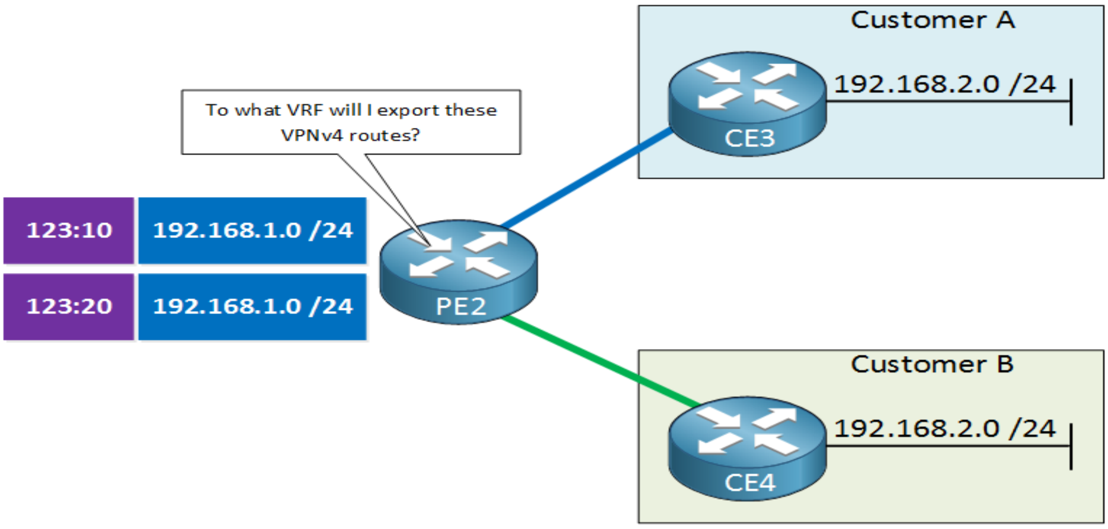


Fig 21. Route target example

Our PE2 router has learned the two VPNv4 routes, one for each customer. You might think that the PE2 router will automatically export each VPNv4 route in the correct customer VRF but that's not going to happen.

The PE2 router will learn 192.168.1.0 /24 from the PE1 router but it has no clue to what customer it will belong. There is no way to differentiate if something belongs to customer A or B.

What we need is something to make all prefixes that we learn unique.

The route target's job is to tell the PE routers what VPN a route actually belongs to.

The route target is also 64 bits but it is an extended community that is sent with the BGP updates. This means that BGP peers must have support for communities. The import decides which routes to import based on the RT and export decides which community is sent with the routes that are exported from the VRF

5.5 VPN Label

The VPN label is to determine what VPN a packet belongs to. But hang on, surely that’s what the RT is for? No. The RT is for the control plane, while the VPN label is for the data plane.

So with L3VPNs we have two labels. The top label is the transport label and the bottom label is the VPN label. PHP will pop the transport label off the second to last router, but the VPN label will only be popped by the actual PE.

5.6 Control Plane

The control plane is the component to a router that focuses on how that one individual box interacts with its neighbors with state exchange. The Routing Information (data) Base (RIB) and Label Information Base (LIB) are processed in software and used to populate FIB (forwarding information base) and the LFIB. Vendors can implement these in different fashions on how those tables are partitioned between multiple routing instances. For example, a router has a BGP and OSPF adjacencies, those routing protocols have different algorithms to determine what a chosen path to a network would be. Building the topology or global view as that particular router sees it from its point of view. That is fairly important to recognize that its “global view” is from its perspective of either the IGP or EGP.

The Control plane feeds the forwarding/data plane with what it needs to create its forwarding tables and updates topology changes as they occur. Those are pretty low even in large networks single to at most I would speculate double digit per second changes. This is the reason the control plane can often be thought of as the “slow path” in legacy route once switch many packet switching architectures. A list of functions performed in traditional routing engines/route processors are the following:

- Allocates resources to the forwarding engine/plane.
- Routing state
- ARP handling is always processed by general purpose processor located in the routing engine.
- Security functions to secure the control plane access. Telnet, ssh, AAA etc.
- Establishes and maintains management sessions, such as Telnet connections
- Routing state to neighboring network elements.
- Vendor and platform specific stacking, clustering, pairing etc.

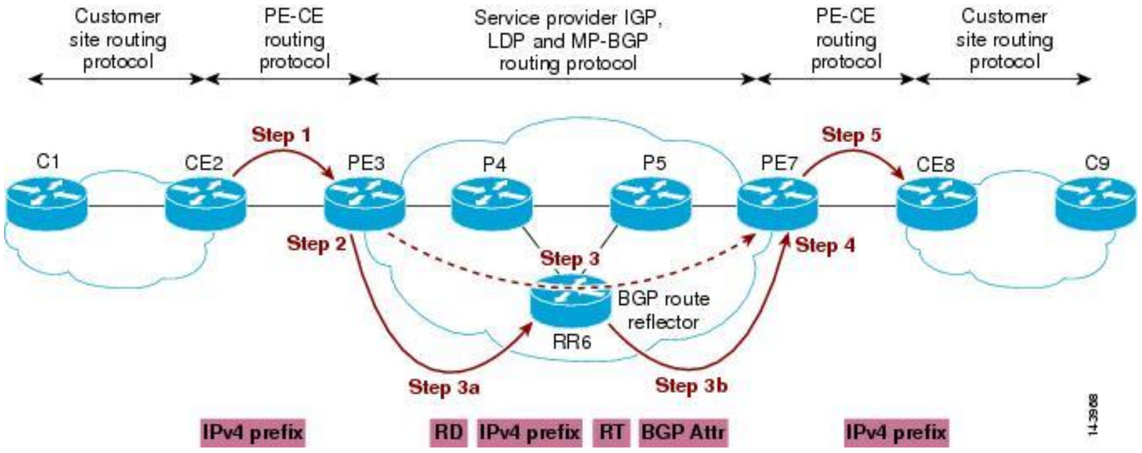


Fig 21 Layer 3 VPN example between two Customer sites

Case study – Layer 3 vpn configuration

During this part of the case study we will configure layer3 vpn and connected branches of customer which are located in different geographical location. Customer routers CE1, CE2, CE3 and CE4 will communicate and exchange routes across provider’s network. In this case OSPF is used as routing protocol between customer routers CE1, CE2, CE3 and PE routers and BGP will be used between CE4 and PE2.

First step during the configuration should be making sure PE1, PE2 and PE3 routers are able to exchange IPv4 and IPv4-vpn prefixes. This is achieved by configuring inet unicast and inet-vpn unicast families “signaling”.

PE1 configuration

```
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 172.16.1.1
set protocols bgp group iBGP family inet unicast
set protocols bgp group iBGP family inet-vpn unicast
set protocols bgp group iBGP neighbor 172.16.1.2
set protocols bgp group iBGP neighbor 172.16.1.3
```

PE2 configuration

```
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 172.16.1.2
set protocols bgp group iBGP family inet unicast
set protocols bgp group iBGP family inet-vpn unicast
set protocols bgp group iBGP neighbor 172.16.1.1
set protocols bgp group iBGP neighbor 172.16.1.3
```

PE3 configuration

```
set protocols bgp group iBGP type internal
set protocols bgp group iBGP local-address 172.16.1.3
set protocols bgp group iBGP family inet unicast
set protocols bgp group iBGP family inet-vpn unicast
set protocols bgp group iBGP neighbor 172.16.1.2
set protocols bgp group iBGP neighbor 172.16.1.1
```

Second part is to configure OSPF/BGP neighborship between PE routers and CE routers.

Configuring vpn instances;

PE1

```
set routing-instances CE1 instance-type vrf
set routing-instances CE1 interface ge-0/0/0.0
set routing-instances CE1 route-distinguisher 172.16.1.3:1
set routing-instances CE1 vrf-target target:65001:1
set routing-instances CE1 protocols ospf area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
set routing-options autonomous-system 65001
```

PE2

```
set routing-instances CE2 instance-type vrf
set routing-instances CE2 interface ge-0/0/1.0
set routing-instances CE2 route-distinguisher 172.16.1.3:1
set routing-instances CE2 vrf-target target:65001:1
set routing-instances CE2 protocols ospf area 0.0.0.0 interface ge-0/0/1.0 interface-type p2p
set routing-options autonomous-system 65001
```

```
set routing-instances CE4 instance-type vrf
set routing-instances CE4 interface ge-0/0/1.0
set routing-instances CE4 route-distinguisher 172.16.1.3:1
set routing-instances CE4 vrf-target target:65001:1
set routing-instances CE4 protocols bgp group CE4 neighbor 10.0.0.6 peer-as 65002
set routing-options autonomous-system 65001
```

PE3

```
set routing-instances CE3 instance-type vrf
set routing-instances CE3 interface ge-0/0/1.0
set routing-instances CE3 route-distinguisher 172.16.1.3:1
set routing-instances CE3 vrf-target target:65001:1
set routing-instances CE3 protocols ospf area 0.0.0.0 interface ge-0/0/1.0 interface-type p2p
set routing-options autonomous-system 65001
```

CE1 configuration:

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/31
set interfaces lo0 unit 10 family inet address 1.1.1.1/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
```

CE2 configuration:

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.2/31
set interfaces lo0 unit 10 family inet address 2.2.2.2/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
```

CE3 configuration:

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.4/31
set interfaces lo0 unit 10 family inet address 3.3.3.3/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
```

CE4 configuration:

```
set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.6/31
set interfaces lo0 unit 10 family inet address 4.4.4.4/32
set protocols ospf area 0.0.0.0 interface ge-0/0/0.0 interface-type p2p
```

VPN routes exchanged between PE neighbors:

```
amehmeti@PE1# run show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                 0         0         0         0         0         0
bgp.l3vpn.0
                 4         4         0         0         0         0
Peer                         AS          InPkt        OutPkt       OutQ      Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
172.16.1.2         65001      26100      26111       0       0      1w1d4h Establ
```

```
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE1.inet.0: 2/2/2/0
172.16.1.3         65001     26106     26112      0      0     1w1d4h Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE1.inet.0: 2/2/2/0
```

```
amehmeti@PE2# run show bgp summary
Groups: 2 Peers: 3 Down peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       0          0          0          0          0          0
bgp.l3vpn.0
                       4          4          0          0          0          0
Peer                                    AS           InPkt         OutPkt       OutQ      Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.6              65002          0          0       0       0     3:14:02 Active
172.16.1.1            65001      26158      26144       0       0      1w1d4h Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE4.inet.0: 2/2/2/0
  CE2.inet.0: 2/2/2/0
172.16.1.3            65001      26149      26145       0       0      1w1d4h Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE4.inet.0: 2/2/2/0
  CE2.inet.0: 2/2/2/0
```

```
amehmeti@PE3# run show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       0          0          0          0          0          0
bgp.l3vpn.0
                       4          4          0          0          0          0
Peer                                    AS           InPkt         OutPkt       OutQ      Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
172.16.1.1            65001      26159      26149       0       0      1w1d4h Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE3.inet.0: 2/2/2/0
```

```
172.16.1.2          65001      26145      26148       0       0      1w1d4h Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 2/2/2/0
  CE3.inet.0: 2/2/2/0
```

Redistributing BGP routes to CE1, CE2 and CE3:

Same config on all three routers:

```
amehmeti@PE1r# set routing-instances CE2 protocols ospf export BGP-TO-OSPF
amehmeti@PE1# set policy-options community VP-routes members target:65001:1

set policy-options policy-statement BGP-TO-OSPF term 1 from protocol bgp
set policy-options policy-statement BGP-TO-OSPF term 1 from community VP-routes
set policy-options policy-statement BGP-TO-OSPF term 1 then accept
set policy-options community VP-routes members target:65001:1
```

For the communication between site CE2 and CE4 also of OSPF routes and direct routes of routing instance CE2 should be distributed to instance CE4 and vice versa.

6. MPLS-Based Layer 2 VPNs

In an MPLS-based Layer 2 VPN, traffic is forwarded by the customer’s customer edge (CE) switch (or router) to the service provider’s provider edge (PE) switch in a Layer 2 format. It is carried by MPLS over the service provider’s network and then converted back to Layer 2 format at the receiving site.

On a Layer 2 VPN, routing occurs on the customer’s switches, typically on the CE switch. The CE switch connected to a service provider on a Layer 2 VPN must select the appropriate circuit on which to send traffic. The PE switch receiving the traffic sends it across the service provider’s network to the PE switch connected to the receiving site. The PE switches do not store or process the customer’s routes; the switches must be configured to send data to the appropriate tunnel.

For a Layer 2 VPN, customers must configure their own switches to carry all Layer 3 traffic. The service provider must detect only how much traffic the Layer 2 VPN will need to carry. The service provider's switches carry traffic between the customer's sites using Layer 2 VPN interfaces. The VPN topology is determined by policies configured on the PE switches.

Customers must know only which VPN interfaces connect to which of their own sites. Figure 18 illustrates a Layer 2 VPN in which each site has a VPN interface linked to each of the other customer sites. In a full-mesh topology between all three sites, each site requires two logical interfaces (one for each of the other CE routers or switches), although only one physical link is needed to connect each PE switch to each CE router or switch.

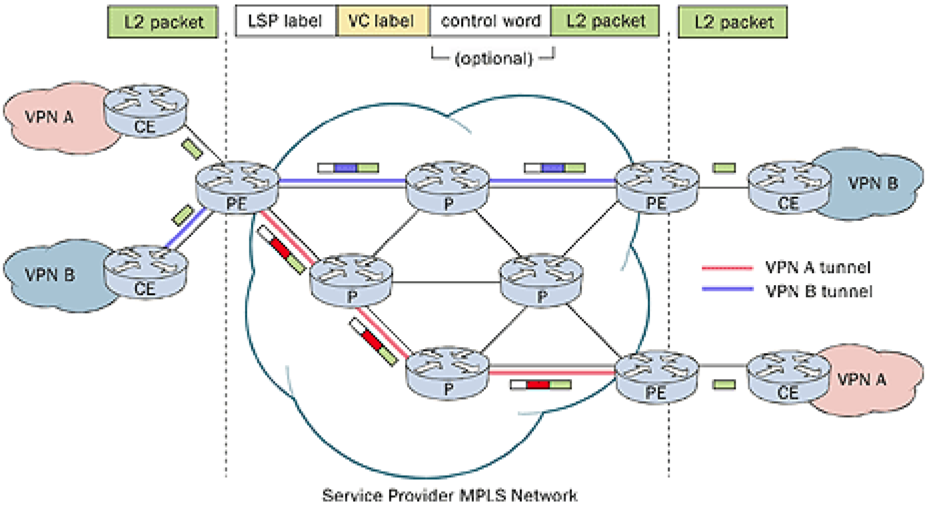


Fig 22 layer 2 VPN exampled

6.1 THE BUSINESS DRIVERS

Native Layer 2 services have existed for several years, based on Frame Relay or ATM. Often these services are used by an enterprise to build its corporate Layer 2 VPN by interconnecting its LANs over a wide area. Service providers can offer near global reach, either directly or through interconnection agreements with partners. The services are a valuable source of revenue to service providers, at the time of writing far outstripping revenues from IP services. In these networks, customer sites are interconnected at Layer 2, sometimes in a full mesh but more typically in a hub-and-spoke topology. The role of the service provider is to transport the ATM cells or Frame Relay frames over the wide area, at an agreed bit-rate for each circuit. As well as being used to carry general LAN interconnection traffic, these services, especially in the ATM case, are sometimes used to carry traffic requiring more stringent SLAs from the network.

In many cases, a service provider can migrate these services to an MPLS network while retaining the same connectivity, as far as the customer is concerned, and maintaining similar service characteristics. In these cases, the presentation to the customer is still over ATM or Frame Relay and a similar service-level agreement (SLA) is offered. For example, in the Frame Relay case, a CIR (committed information rate) is agreed for each circuit and SLA is defined for parameters such as packet loss, latency and delay
Variation.
Migrating these services to an MPLS network saves the service provider capital and operational expenses compared to running separate networks for Layer 2 connectivity and Layer 3 connectivity. Also one of the schemes discussed later greatly reduces the operational burden of provisioning Layer 2 connections within the service provider part of the network, especially in cases where a high degree of meshing is used between customer sites, which leads to a further saving in operational costs. Another growing application of Layer 2 transport over MPLS is Ethernet services, in which a customer's Ethernet frames are transported between the customer's sites over the service provider's MPLS network. The appeal to the end customer is that Ethernet is the standard Layer 2 protocol used within the enterprise and hence is familiar to the corporate IT staff. Using Ethernet to interconnect their sites over the wide area is a natural extension of the use of Ethernet within their premises. In many cases where
Customers have been using ATM or Frame Relay services for LAN interconnection,
there is no fundamental reason why ATM or Frame Relay should be used as the interconnectivity method. Ethernet has the attraction that it is more flexible in terms of

access rates – the service provider can offer, for example, a 100 Mbps Ethernet tail that is rate-limited to the level paid for by the customer. This allows for smoother upgrades in access speed than having, for example, to change from an E1/T1 access circuit to an E3/T3 access circuit. These factors, along with the fact that Ethernet-base equipment tends to be less expensive than ATM or Frame Relay equipment, by

virtue of volume, mean that in some cases a customer might migrate from a native ATM or Frame Relay based service to an Ethernet service in order to reduce costs. Similarly, enterprises using leased-line services for LAN interconnection can reduce costs by switching to Ethernet services.

6.2 COMPARISON OF LAYER 2 VPNs AND LAYER 3 VPNs

The introduction to the Layer 3 VPN is discussed the two main models that exist for VPN connectivity: the overlay model and the peer model. BGP/MPLS-based Layer 3 VPNs fall within the peer model. In contrast, when an enterprise builds a Layer 2 VPN, by buying Layer 2 transport services from the service provider they are building an overlay network. Hence the differences between Layer 2 and Layer 3 VPNs are as follows:

1. In the Layer 2 case, no routing interaction occurs between the customer and service provider. In the L3VPN case, the CE and PE router can exchange routes.

2. In the Layer 2 case, the customer can run any type of Layer 3 protocol between sites. The SP network is simply transporting Layer 2 frames and hence is unaware of the Layer 3 protocol that is in use. 3. Multiple (logical) interfaces between each CE and the corresponding PE are required in the Layer 2 case, one per remote CE that each CE needs to connect to. For example, if the CE routers are fully meshed and there are 10 CE routers in total, each CE needs nine interfaces (e.g. DLCIs, VCs or VLANs, depending on the media type) to the PE, each leading to one of the remote CE routers. In the Layer 3 VPN case, one connection between each CE and the local PE is sufficient as the PE is responsible for routing the traffic towards the appropriate egress CE. For some customers, L3VPN is the better choice, for others L2VPN, depending on what

protocols need to be carried and the degree to which the customer wishes to do their own routing or to outsource it to the service provider. Hence, in order to address the widest possible market, many service providers offer both Layer 3 and Layer 2 services over their MPLS

6.3 PRINCIPLES OF LAYER 2 TRANSPORT OVER MPLS

There are two main approaches to Layer 2 transport over MPLS:

One involving LDP signaling [MRT-TRS] [PWE3-CON] and the other based on BGP signaling [KOM-BGP]. In the forwarding plane, these approaches are the same, in terms of how Layer 2 frames are encapsulated for transport across the MPLS network. However, the two approaches differ significantly in the control plane. A single point-to-point Layer 2 connection provided over an MPLS network is sometimes called a pseudowire, to convey the principle that as far as possible the MPLS network should be invisible to the end customer, in such a way that the two CEs interconnected by the pseudowire appear to be directly connected back to back. An MPLS-based L2VPN is composed of a collection of pseudowires that interconnect a customer's CEs in different locations, in a topology chosen by the customer, for example a full-mesh or hub-and-spoke arrangement.

One of the problems with traditional Layer 2 VPNs is the administrative burden of adding a new site to an existing VPN, and the associated lead-times. If the sites are fully meshed, when a new site is introduced a new circuit must be provisioned between the new site and every other site in the network, and hence extra configuration at every site in the network is required. Indeed, often this administrative burden has forced customers to adopt a hub-and-spoke arrangement. Later we will show how autodiscovery of sites using BGP greatly reduces the administrative overhead associated with traditional Layer 2 VPNs by making it much easier to add new sites to an existing mesh.

Examples of Layer 2 protocol types that can be carried over an MPLS network are as follows:

1. ATM. Two main modes exist: a mode in which AAL5 PDUs are transported on the pseudowire and a mode in which ATM cells are transported on the pseudowire. In the latter case, the cells could belong to any AAL type, since the AAL PDUs are not reassembled by the MPLS network.

2. Ethernet. The mapping of traffic into a pseudowire can be on a per-VLAN or on a per-port basis. In the per-VLAN case, if an Ethernet connection between the customer CE router and the service provider's PE router contains multiple VLANs, each VLAN can be mapped to a different pseudowire for transport to a different remote CE.

3. Frame Relay. The mapping of traffic into a pseudowire can be on a per port basis or on a per-DLCI basis. In the per-DLCI case, if a Frame Relay connection between the customer CE router and the service provider's PE router contains multiple DLCIs, each DLCI can be mapped to a different pseudowire for transport to a different remote CE.

6.4 Ethernet

Two modes of Ethernet transport [PWE3-ETH] exist, one in which the mapping to pseudowires across the core is on a per-VLAN basis and another in which an entire Ethernet port, which may contain multiple VLANs, is mapped to a single pseudowire. The use of the Control Word is optional, but if used there is a 16-bit sequence number that can be used if required. The FCS is stripped off at the ingress PE and regenerated by the egress PE. The Control Word in the Ethernet case is generally regarded as less useful than in the ATM or Frame Relay cases.

6.5 CONTROL PLANE OPERATION

Let us see how the control plane for Layer 2 transport operates. We will examine the LDP-based scheme [MRT-TRS] and the BGP-based scheme [KOM-BGP]. Both approaches have the following characteristics in common:

1. A means for a PE, when forwarding traffic from a local CE via a remote

PE to a remote CE, to know the value of the VPN label (inner label) that
the remote PE expects
.

2. A means for signaling characteristics of the pseudowire, such as media
type and MTU. This provides a means to detect whether each end of a
pseudowire are configured in a consistent manner or not.

3. An assumption that the pseudowire formed is bidirectional.1 Hence, if
there is a problem with transport in one direction, forwarding is not
Allowed to occur in the opposite direction.

4. A means for a PE to indicate to remote PE(s) that there is a problem
with connectivity, e.g. if the link to a CE goes down.

The two schemes differ significantly in the way in which a PE knows which remote PE(s) it needs to build pseudowires to. In the original LDP based scheme, this information had to be manually configured on the PEs. The BGP scheme, in contrast, has in-built autodiscovery properties, so this manual configuration is not required. The original LDP scheme was later modified in order to also avoid this manual configuration, by using information discovered by some means external to LDP. One option for the autodiscovery aspect is to use BGP.

7.VPLS

7.1. Introduction

Various implementations of VPLS enable the delivery of multipoint Ethernet services. VPLS is not only being used by service providers to offer transparent LAN service to enterprise customers, but with the emergence of metro Ethernet networks, VPLS is also being used as an infrastructure technology. Service providers are showing significant interest as measured by their VPLS deployments; many are now offering inter-provider services. This interest has been fueled by the fact major multiple service operators (MSOs) and large service providers are increasingly targeting small- and medium-sized businesses (SMB), as well as large enterprises. Consequently, the growing demand for VPLS requires the underlying enabling infrastructure to scale to support larger numbers of concurrent VPLS instances with multiple sites spread across geographically dispersed regions.

This document specifies in detail the very mechanisms that allow the scalable rollout of VPLS, as well as the enabling mechanisms that allow the interworking of LDP- and BGP-based VPLS.

7.2 VPLS Overview

A VPLS network is a Layer 2 multipoint VPN that emulates LAN services across a WAN. VPLS enables service providers to interconnect several customer sites (each being a LAN segment) over a packet-switched network, effectively making all the customer LAN segments behave as one single LAN. With VPLS, no routing interaction occurs between the customer and service provider, and the customer can run any type of Layer 3 protocol between sites.

The IETF Layer 2 VPN Working Group has produced two separate VPLS standards, documented in RFC 4761 and RFC 4762 (see Kompella and Rekhter, Jan. 2007, and Lasserre and Kompella, Jan. 2007). These two RFCs define almost identical approaches with respect to the VPLS data plane, but specify significantly different approaches to implementing the VPLS control planes.

7.3. VPLS Control Plane

The VPLS control plane has two primary functions: autodiscovery and signaling.

Discovery refers to the process of finding all PE routers that participate in a given VPLS instance. A PE router can be configured with the identities of all the other PE routers in a given VPLS instance, or the PE router can use a protocol to discover the other PE routers. The latter method is called autodiscovery. After discovery occurs, each pair of PE routers in a VPLS network must be able to establish pseudowires to each other, and in the event of membership change, the PE router must be able to tear down the established pseudowires. This process is known as signaling. Signaling is also used to transmit certain characteristics of the pseudowire that a PE router sets up for a given VPLS.

7.4. BGP-VPLS Control Plane

The BGP-VPLS control plane, as defined by RFC 4761, is similar to that for Layer 2 and Layer 3. It defines a means for a PE router to discover which remote PE routers are members of a given VPLS (autodiscovery), and for a PE router to know which pseudowire label a given remote PE router will use when sending the data to the local PE router (signaling). With the BGP-VPLS control plane, BGP carries enough information to provide the autodiscovery and signaling functions simultaneously.

As in the BGP scheme for Layer 2 and Layer 3 VPNs, a route target is configured on each PE router for each VPLS present on the PE router. The route target is the same for a particular VPLS across all PE routers and is used to identify the VPLS instance to which an incoming BGP message pertains. For each VPLS on each PE router, an identifier is configured, known as a site identifier. Each PE router involved in a particular VPLS must be configured with a unique site identifier. The site identifier is same as the Virtual Edge Identifier (VEID) referred to in RFC 4761.

A label block is a set of demultiplexor labels used to reach a given VPLS site within a set of remote sites. The PE router uses a label block to send a single common update message to establish a pseudowire with multiple PE routers, instead of having to send an individual message to each PE router.

5.4 LDP-VPLS Control Plane

In contrast to the BGP-VPLS control plane, the LDP-VPLS control plane provides only signaling, but not autodiscovery. In this control plane, LDP is used to signal the pseudowires that are used to interconnect the VPLS instances of a given customer on the PE routers. The LDP signaling scheme for VPLS is similar to the LDP scheme for point-to-point Layer 2 connections. In the absence of an autodiscovery mechanism, the identities of all the remote PE routers that are part of a VPLS instance must be configured on each PE router.

The virtual circuit identifier (VCID), which is in the point-to-point Layer 2 connection used to identify a specific pseudowire, is configured to be the same for a particular

VPLS instance on all PE routers. Hence, the VCID enables a PE router to identify the VPLS instance to which the LDP message refers. LDP VPLS defines the hierarchical VPLS (H-VPLS) scheme in which, instead of all PE routers being fully meshed with LDP sessions, a two-level hierarchy is created involving hub PE routers and spoke PE routers. The hub PE routers are fully meshed with LDP sessions, whereas a spoke PE router has a pseudowire only to a single hub PE

router or to multiple hub PE routers for redundancy. Spoke pseudowires can be implemented using any Layer 2 tunneling technology.

5.5. Forwarding Plane

Forwarding plane procedures, at least for unicast and to some extent for multicast, are the same for both BGP VPLS and LDP VPLS. For each VPLS, a PE VPLS data plane functions as a learning bridge and supports all the standard bridge operations, such as MAC address learning, aging, and flooding. All the pseudowires established by BGP or LDP signaling and the local customer edge (CE) router ports of a VPLS instance constitute the logical ports of a bridge domain.

A MAC forwarding table is created for each VPLS instance on a PE router. This table is populated using a source MAC address learning function and is used to forward unicast VPLS traffic based on the destination MAC address of the received frame.

VPLS packets received over a pseudowire are not forwarded to any other pseudowire, but rather are forwarded only on the attached CE router circuit. This behavior has been called split-horizon forwarding and is a consequence of the

PE routers being logically fully meshed in the data plane for each VPLS instance. The use of a full mesh combined with split-horizon forwarding avoids Layer 2 loops and is the VPLS alternative to Spanning Tree Protocol (STP) within the service provider network.

7.5. Scaling Characteristics of LDP VPLS and BGP VPLS

There are two distinct VPLS control planes: LDP based and BGP based, both of which have been standardized by the IETF. This section compares the scaling aspects of the signaling and discovery mechanisms of these two control planes.

7.6. Full-Mesh Connectivity

To enable VPLS, all PE routers connected to common VPLS customers must be able to exchange VPLS signaling information amongst themselves. As the number of PE routers in the network increases, scaling this signaling component of the VPLS control plane becomes essential. For LDP-VPLS signaling, the exchange of VPLS signaling information is accomplished by setting up a full mesh of targeted LDP sessions between each pair of PE routers that have at least one VPLS in common .As the size of the VPLS network grows, the number of LDP targeted sessions increases exponentially on the order of O(N^2), where N is the number of LDP-VPLS PE routers in the network. Maintenance of all these LDP sessions creates an additional load on the control plane of PE routers in the VPLS network. The operational challenge resulting from the O(N^2) increase in LDP sessions becomes even more noticeable when a service provider authenticates the sessions using Message Digest 5 (MD5) because MD5 keys must be configured on each end of every LDP session. Adding a new PE router or deleting an existing one becomes a cumbersome task because the configuration on each of the PE routers in the network must be modified. Unlike in LDP VPLS, the exchange of VPLS signaling information in BGP VPLS does not require a full mesh of control sessions among all PE routers. Instead, the BGP-VPLS control plane, including its signaling component, can use a route reflector (RR) hierarchy, in which only the route reflectors are fully meshed (Figure 2). Each BGP router then establishes a BGP session to one or more route reflectors. Using route reflectors also makes the provisioning task of adding or deleting a PE router simpler because only the BGP peering with the route reflector needs to be changed.

Route reflectors are a proven technology that is used extensively in networks in which BGP is deployed for Internet routing or for other types of VPNs. Furthermore, BGP-VPLS route reflectors can be placed anywhere in the network; that is, they do not need to be on the data path of the VPLS domains they host. This arrangement offers flexibility in deploying new route reflectors when needed for even greater scaling. LDP VPLS signaling has no mechanism similar to the route reflector hierarchy that can eliminate the full mesh of targeted LDP sessions. H-VPLS tries, to some extent, to mitigate the full-mesh requirement by creating a two-level hierarchy of hub and spoke devices. However, this attempt to scale the control plane changes the data plane behavior and has a significant negative impact on the data plane. In contrast, because the BGP-VPLS route reflector technique is purely a control plane technique, it does not change the forwarding path of VPLS traffic in any way, and thus has no impact on the data plane.

7.7. Flooding and Broadcasting

Flooding and broadcasting operations using ingress replication in a large VPLS topology can be both resource intensive and bandwidth inefficient. These drawbacks are especially apparent on the first hop link from the PE router, which may have to bear the full burden of the replication

Point-to-multipoint (P2MP) label-switched paths (LSPs) introduce multicast to MPLS and enable several new services, such as the distribution of IP-based television (IPTV). Enhancements in BGP VPLS enable the use of a P2MP

LSP VPLS-multicast for efficient and scalable flooding and broadcasting of VPLS traffic. Junos operating system supports this functionality using RSVP-TE P2MP LSP.While in principle LDP VPLS can also use P2MP LSP to scale the data plane, in practice no vendor has yet integrated these two technologies.

7.8 Exchanging Pseudowire Signaling State

In both LDP VPLS and BGP VPLS, for each VPLS customer, signaling updates are exchanged between each pair of member PE routers to create full-mesh connectivity in the data plane. The amount of signaling state that each PE router needs to maintain and to exchange with other PE routers increases as VPLS expands, both in terms of the number of VPLS customers and the number of member PE routers for each VPLS customer. A single LDP-VPLS signaling update establishes only one pseudowire with one other PE router. In contrast, a single BGP-VPLS signaling update establishes pseudowires with multiple PE router peers. Consequently, in large VPLS networks, BGP VPLS requires far fewer signaling updates and therefore many fewer control plane messages than

LDP VPLS to establish a full mesh of pseudowires between pairs of PE routers. Having fewer control plane messages reduces the control plane overhead on each PE router.

7.9. VPLS Multihoming Overview

Virtual private LAN service (VPLS) multihoming enables you to connect a customer site to two or more PE routers to provide redundant connectivity. A redundant PE router can provide network service to the customer site as soon as a failure is detected. VPLS multihoming helps to maintain VPLS service and traffic forwarding to and from the multihomed site in the event of the following types of network failures:

- PE router to CE device link failure
- PE router failure
- MPLS-reachability failure between the local PE router and a remote PE router

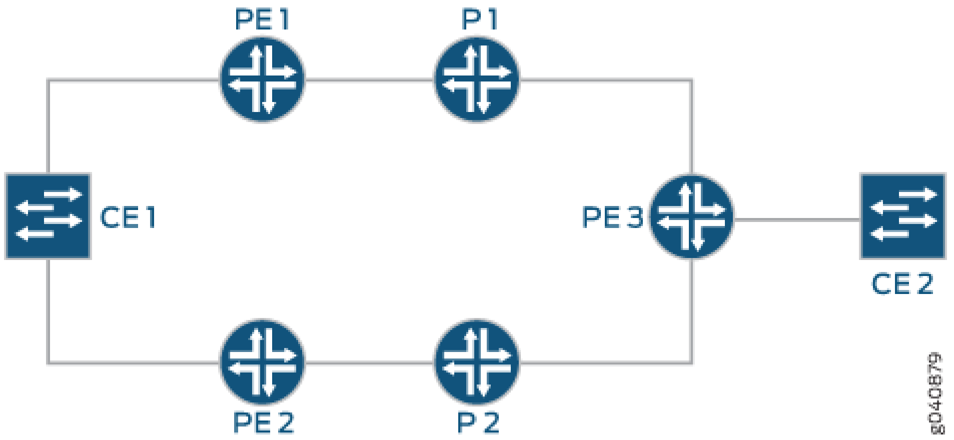


Fig 23. Multihomed VPLS example

Figure 23 illustrates how a CE device could be multihomed to two PE routers. Device CE1 is multihomed to Routers PE1 and PE2. Device CE2 has two potential paths to reach Device CE1, but only one path is active at any one time

Multihomed PE routers advertise network layer reachability information (NLRI) for the multihomed site to the other PE routers in the VPLS network. The NLRI includes the site ID for the multihomed PE routers. For all of the PE routers multihomed to the same CE device, you need to configure the same site ID. The remote VPLS PE routers use the site ID to determine where to forward traffic addressed to the customer site. To avoid route collisions, the site ID shared by the multihomed PE routers must be different than the site IDs configured on the remote PE routers in the VPLS network.

Although you configure the same site ID for each of the PE routers multihomed to the same CE device, you can configure unique values for other parameters, such as the route distinguisher. These values help to determine which multihomed PE router is selected as the designated VE device to be used to reach the customer site.

Remote PE routers in the VPLS network need to determine which of the multihomed PE routers should forward traffic to reach the CE device. To make this determination, remote PE routers use the VPLS path-selection process to select one of the multihomed PE routers based on its NLRI advertisement. Because remote PE routers pick only one of the NLRI advertisements, it establishes a pseudowire to only one of the multihomed PE routers, the PE router that originated the winning advertisement. This prevents multiple paths from being created between sites in the network, preventing the formation of Layer 2 loops. If the selected PE router fails, all PE routers in the network automatically switch to the backup PE router and establish new pseudowires to it.

Case study Part Three- VPLS implementation ISP connecting customer

As mentioned there is VPLS BGP based and LDP based. In our case study we will use BGP based VPLS for connecting CE1,CE2 and CE3.

In all three PE routes l2vpn signaling must be configured as below:

```
amehmeti@PE1# set protocols bgp group iBGP family l2vpn signaling
```

PE1

```
set routing-instances CE1 instance-type vpls
set routing-instances CE1 interface ge-0/0/0.0
set routing-instances CE1 route-distinguisher 172.16.1.1:1
set routing-instances CE1 vrf-target target:65001:1
set routing-instances CE1 protocols vpls site-range 10
set routing-instances CE1 protocols vpls no-tunnel-services
set routing-instances CE1 protocols vpls site CE1 site-identifier 1
set routing-options autonomous-system 65001
```

PE2

```
set routing-instances CE2 instance-type vpls
set routing-instances CE2 interface ge-0/0/0.0
set routing-instances CE2 route-distinguisher 172.16.1.2:2
set routing-instances CE2 vrf-target target:65001:1
set routing-instances CE2 protocols vpls site-range 10
set routing-instances CE2 protocols vpls no-tunnel-services
set routing-instances CE2 protocols vpls site CE2 site-identifier 2
```

PE3

```
set routing-instances CE3 instance-type vpls
set routing-instances CE3 interface ge-0/0/0.0
set routing-instances CE3 route-distinguisher 172.16.1.3:1
set routing-instances CE3 vrf-target target:65001:1
set routing-instances CE3 protocols vpls site-range 10
set routing-instances CE3 protocols vpls no-tunnel-services
set routing-instances CE3 protocols vpls site CE3 site-identifier 3
set routing-options autonomous-system 65001
```

```
amehmeti@PE1# run show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                 0         0         0         0         0         0
bgp.l3vpn.0
                 0         0         0         0         0         0
bgp.l2vpn.0
                 2         2         0         0         0         0
Peer                              AS          InPkt        OutPkt       OutQ      Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
172.16.1.2           65001        19         17       0       0       6:19 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE1.l2vpn.0: 1/1/1/0
172.16.1.3           65001        20         18       0       0       6:19 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE1.l2vpn.0: 1/1/1/0
```

```
amehmeti@PE2# run show bgp summary
Groups: 2 Peers: 3 Down peers: 1
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                 0         0         0         0         0         0
bgp.l3vpn.0
                 0         0         0         0         0         0
bgp.l2vpn.0
                 2         2         0         0         0         0
```

```
Peer                     AS        InPkt       OutPkt      OutQ     Flaps  Last  Up/Dwn
State|#Active/Received/Accepted/Damped...
10.0.0.6          65002        0         0       0       0     4:17:45 Active
172.16.1.1          65001        20        20       0       0        6:58 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE2.l2vpn.0: 1/1/1/0
172.16.1.3          65001        20        18       0       0        6:54 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE2.l2vpn.0: 1/1/1/0
```

```
amehmeti@PE3# run show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0
                       0          0          0          0          0          0
bgp.l3vpn.0
                       0          0          0          0          0          0
bgp.l2vpn.0
                       2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
172.16.1.1            65001         21         20       0       0        7:23 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE3.l2vpn.0: 1/1/1/0
172.16.1.2            65001         20         19       0       0        7:19 Establ
  inet.0: 0/0/0/0
  bgp.l3vpn.0: 0/0/0/0
  bgp.l2vpn.0: 1/1/1/0
  CE3.l2vpn.0: 1/1/1/0
```

Verifying that vpls is working:

```
amehmeti@PE1# run show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
```

```
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: CE1
 Local site: CE1 (1)
  connection-site           Type  St     Time last up          # Up trans
  2                         rmt   Up     Jun  5 23:07:05 2016           1
   Remote PE: 172.16.1.2, Negotiated control-word: No
   Incoming label: 262146, Outgoing label: 262145
   Local interface: lsi.34603009, Status: Up, Encapsulation: VPLS
    Description: Intf - vpls CE1 local site 1 remote site 2
  3                         rmt   Up     Jun  5 23:07:05 2016           1
   Remote PE: 172.16.1.3, Negotiated control-word: No
   Incoming label: 262147, Outgoing label: 262145
   Local interface: lsi.34603008, Status: Up, Encapsulation: VPLS
    Description: Intf - vpls CE1 local site 1 remote site 3


amehmeti@PE2# run show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
```

```
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: CE2
  Local site: CE2 (2)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jun  5 23:07:05 2016           1
      Remote PE: 172.16.1.1, Negotiated control-word: No
      Incoming label: 262145, Outgoing label: 262146
      Local interface: lsi.84934656, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls CE2 local site 2 remote site 1
    3                         rmt   Up     Jun  5 23:07:09 2016           1
      Remote PE: 172.16.1.3, Negotiated control-word: No
      Incoming label: 262147, Outgoing label: 262146
      Local interface: lsi.84934657, Status: Up, Encapsulation: VPLS
    Description: Intf - vpls CE2 local site 2 remote site 3


amehmeti@PE3# run show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not available
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor
LB -- Local site not best-site   RB -- Remote site not best-site
VM -- VLAN ID mismatch

Legend for interface status
Up -- operational
Dn -- down

Instance: CE3
  Local site: CE3 (3)
    connection-site           Type  St     Time last up          # Up trans
    1                         rmt   Up     Jun  5 23:07:05 2016           1
      Remote PE: 172.16.1.1, Negotiated control-word: No
      Incoming label: 262145, Outgoing label: 262147
      Local interface: lsi.51380224, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls CE3 local site 3 remote site 1
    2                         rmt   Up     Jun  5 23:07:09 2016           1
      Remote PE: 172.16.1.2, Negotiated control-word: No
      Incoming label: 262146, Outgoing label: 262147
      Local interface: lsi.51380225, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls CE3 local site 3 remote site 2
```

Appendixes

8. Software Defined Networking

8.1 Definition

Software Defined Networking (SDN) The goal of Software-Defined Networking is to enable cloud and network engineers and administrators to respond quickly to changing business requirements via a centralized control console. SDN encompasses multiple kinds of network technologies designed to make the network more flexible and agile to support the virtualized server and storage infrastructure of the modern data center and Software defined networking was originally defined an approach to designing, building, and managing networks that separates the network's control (brains) and forwarding (muscle) planes enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

How Does Software-Defined Networking or SDN Work?

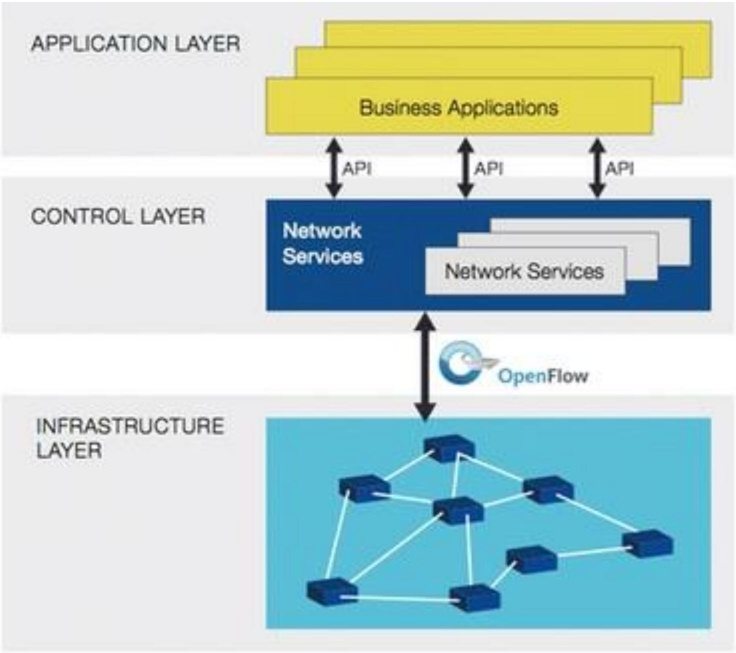
Software-defined networking providers offer a wide selection of competing architectures, but at its most simple, the Software Defined Networking method centralizes control of the network by separating the control logic to off-device computer resources. All SDN models have some version of an SDN Controller, as well as southbound APIs and northbound APIs:

Controllers: The "brains" of the network, SDN Controllers offer a centralized view of the overall network, and enable network administrators to dictate to the underlying systems (like switches and routers) how the forwarding plane should handle network traffic.

Southbound APIs: Software-defined networking uses southbound APIs to relay information to the switches and routers "below." OpenFlow, considered the first standard in SDN, was the original southbound API and remains as one of the most common protocols. Despite some considering OpenFlow and SDN to be one in the same, OpenFlow is merely one piece of the bigger SDN landscape.

Northbound APIs: Software Defined Networking uses northbound APIs to communicates with the applications and business logic “above.” These help network administrators to programmatically shape traffic and deploy services.

8.2 The Software Defined Networking Framework



Software-Defined Networking is Not OpenFlow

Often people point to OpenFlow as being synonymous with software-defined networking, but it is only a single element in the overall SDN architecture. OpenFlow is an open standard for a communications protocol that enables the control plane to interact with the forwarding plane. It must be noted that OpenFlow is not the only protocol available or in development for SDN.

The Benefits of Software Defined Networking

Offering a centralized, programmable network that can dynamically provision so as to address the changing needs of businesses, software-define networking also provides the following benefits:

Directly Programable: Network directly programmable because the control functions are decoupled from forwarding functions which enable the network to be programmatically configured by proprietary or open source automation tools, including OpenStack, Puppet, and Chef.

Centralized Management: Network intelligence is logically centralized in SDN controller software that maintains a global view of the network, which appears to applications and policy engines as a single, logical switch.

Reduce CapEx: Software Defined Networking potentially limits the need to purchase purpose-built, ASIC-based networking hardware, and instead supports pay-as-you-grow models

Reduce OpEX: SDN enables algorithmic control of the network of network elements (such as hardware or software switches / routers that are increasingly programmable, making it easier to design, deploy, manage, and scale networks. The ability to automate provisioning and orchestration optimizes service availability and reliability by reducing overall management time and the chance for human error.

Deliver Agility and Flexibility: Software Defined Networking helps organizations rapidly deploy new applications, services, and infrastructure to quickly meet changing business goals and objectives.

Enable Innovation: SDN enables organizations to create new types of applications, services, and business models that can offer new revenue streams and more value from the network.

9. Reference:

1. *Cisco@2005 - Routing TCPIP, Volume I 2nd Edition By Jeff Doyle CCIE 1919*
2. *CCIE Routing and Switching, CISCO Press 2007, 3rd edition By Wendell Odom, CCIE No.1624,Rus Healy CCIE No. 15025 and contributing author: Naren Mehta CCIE No.9797*
3. *MPLS-Enabled Applications: Emerging Developments and New Technologies, 3rd Edition, Ina Minei, Julian Lucek*
4. *MPLS Fundamentals , Luc De Ghein*
5. *MPLS and VPN Architectures, By Ivan Pepelnjak, Jim Guichard*
6. *Cisco Quality of Service Solutions Configuration guide, release 12.2, 170West Tasman Drive San JOSE, CA 95134-1706*
7. *Day One: MPLS for Enterprise Engineers5 Mar 2014 by Darren O'Conno*
8. *Deploying MPLS2 May 2011 by Tim Fiola and Jamie Panagos*